Qualys API
(VM, PC)
User Guide
Version 10.24.1

November 09, 2023
Copyright 2018-2023 by Qualys, Inc. All Rights Reserved.

Qualys and the Qualys logo are registered trademarks of Qualys, Inc. All other trademarks are the property of their respective owners.

Qualys, Inc.
919 E Hillsdale Blvd
4th Floor
Foster City, CA 94404
1 (650) 801 6100
Table of Contents

Preface ................................................................................................................................. 9

Welcome ................................................................................................................................. 10
  API Conventions ............................................................................................................... 10
  Qualys User Account ...................................................................................................... 10
  Qualys API Server URL .................................................................................................. 11
  Making API requests ...................................................................................................... 12
  API Limits ....................................................................................................................... 13
  Tracking API usage by user ......................................................................................... 14
  HTTP Response Headers ............................................................................................... 14
  Activity Log ..................................................................................................................... 17

Authentication to your account ....................................................................................... 18
  What do I need to know? ............................................................................................... 18
  Using Basic HTTP Authentication ............................................................................. 18
  Using Session Based Authentication .......................................................................... 19
  Session Login .................................................................................................................. 22
  Session Logout ................................................................................................................ 24

Scans ..................................................................................................................................... 26
  VM Scans ....................................................................................................................... 27
  VM Scan List .................................................................................................................. 27
  List Last ‘N’ Scan References for a Schedule ............................................................... 30
  Launch VM Scan ............................................................................................................ 33
  Launch VM Scan on EC2 assets ................................................................................... 35
  Manage VM Scans ......................................................................................................... 38
  Compliance Scans .......................................................................................................... 41
  Compliance Scan List ................................................................................................... 42
  List Last ‘N’ Scan References for a Schedule ............................................................... 43
  SCAP Scan List .............................................................................................................. 46
  Launch Compliance Scan ............................................................................................ 48
  Launch Compliance Scan on EC2 assets .................................................................... 49
  Manage Compliance Scans .......................................................................................... 51
  Cloud Perimeter Scans ................................................................................................. 54
  Create/Update Cloud Perimeter Scan ........................................................................ 54
  VM Scan Schedules ....................................................................................................... 57
  PC Scan Schedules ......................................................................................................... 68
  Scan List Parameters ..................................................................................................... 78
  Scan Parameters ............................................................................................................ 80
  Cloud Perimeter Scan Parameters .............................................................................. 85
  Scan Schedule Parameters ........................................................................................... 89
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS Exchange Server</td>
<td>332</td>
</tr>
<tr>
<td>MS SQL Record</td>
<td>337</td>
</tr>
<tr>
<td>MySQL Record</td>
<td>346</td>
</tr>
<tr>
<td>Neo4j Record</td>
<td>353</td>
</tr>
<tr>
<td>Nginx Record</td>
<td>358</td>
</tr>
<tr>
<td>Oracle Record</td>
<td>362</td>
</tr>
<tr>
<td>System created Oracle records</td>
<td>367</td>
</tr>
<tr>
<td>Oracle Listener Record</td>
<td>370</td>
</tr>
<tr>
<td>Oracle WebLogic Server Record</td>
<td>372</td>
</tr>
<tr>
<td>Palo Alto Firewall Record</td>
<td>375</td>
</tr>
<tr>
<td>Pivotal Greenplum Record</td>
<td>379</td>
</tr>
<tr>
<td>PostgreSQL Record</td>
<td>386</td>
</tr>
<tr>
<td>SAP Hana Record</td>
<td>393</td>
</tr>
<tr>
<td>SAP IQ Record</td>
<td>398</td>
</tr>
<tr>
<td>SNMP Record</td>
<td>403</td>
</tr>
<tr>
<td>Sybase Record</td>
<td>408</td>
</tr>
<tr>
<td>Unix Record</td>
<td>414</td>
</tr>
<tr>
<td>Network SSH Record</td>
<td>431</td>
</tr>
<tr>
<td>VMware Record</td>
<td>438</td>
</tr>
<tr>
<td>Windows Record</td>
<td>444</td>
</tr>
<tr>
<td>Oracle HTTP Server Record</td>
<td>452</td>
</tr>
<tr>
<td>vCenter - ESXi Mapping Records</td>
<td>462</td>
</tr>
<tr>
<td>Vault Support</td>
<td>468</td>
</tr>
<tr>
<td>Vault Support matrix</td>
<td>468</td>
</tr>
<tr>
<td>Vault Definition</td>
<td>474</td>
</tr>
<tr>
<td>List Vaults</td>
<td>481</td>
</tr>
<tr>
<td>Manage Vaults</td>
<td>484</td>
</tr>
<tr>
<td>Assets</td>
<td>495</td>
</tr>
<tr>
<td>IP List</td>
<td>496</td>
</tr>
<tr>
<td>Add IPs</td>
<td>499</td>
</tr>
<tr>
<td>Update IPs</td>
<td>501</td>
</tr>
<tr>
<td>Host List</td>
<td>505</td>
</tr>
<tr>
<td>Host Update</td>
<td>518</td>
</tr>
<tr>
<td>Host List Detection</td>
<td>522</td>
</tr>
<tr>
<td>Host List Detection - Normalized Data</td>
<td>544</td>
</tr>
<tr>
<td>Host List Detection - Use Cases</td>
<td>545</td>
</tr>
<tr>
<td>Host List Detection - Best Practices</td>
<td>546</td>
</tr>
<tr>
<td>Excluded Host List</td>
<td>547</td>
</tr>
<tr>
<td>Excluded Hosts Change History</td>
<td>550</td>
</tr>
<tr>
<td>Manage Excluded Hosts</td>
<td>553</td>
</tr>
<tr>
<td>Virtual Host List</td>
<td>557</td>
</tr>
<tr>
<td>Manage Virtual Hosts</td>
<td>558</td>
</tr>
</tbody>
</table>
Preface

Using the Qualys API, third parties can integrate their own applications with Qualys cloud security and compliance solutions using an extensible XML interface. The APIs described in this guide are available to customers using Qualys Cloud Platform (VM, PC).

About Qualys

Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of cloud-based security and compliance solutions. The Qualys Cloud Platform and its integrated apps help businesses simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications.

Founded in 1999, Qualys has established strategic partnerships with leading managed service providers and consulting organizations including Accenture, BT, Cognizant Technology Solutions, Deutsche Telekom, Fujitsu, HCL, HP Enterprise, IBM, Infosys, NTT, Optiv, SecureWorks, Tata Communications, Verizon and Wipro. The company is also a founding member of the Cloud Security Alliance (CSA). For more information, please visit www.qualys.com.

Contact Qualys Support

Qualys is committed to providing you with the most thorough support. Through online documentation, telephone help, and direct email support, Qualys ensures that your questions will be answered in the fastest time possible. We support you 7 days a week, 24 hours a day. Access support information at www.qualys.com/support/.
Welcome

The Qualys API allows third parties to integrate their own applications with Qualys cloud security and compliance solutions using an extensible XML interface. APIs in this user guide are supported using Qualys Cloud Platform (VM, PC).

We recommend you join our Community and subscribe to our API Notifications RSS Feeds for announcements and discussions.

API Conventions

Qualys User Account

Authentication with valid Qualys user account credentials is required for making Qualys API requests to the Qualys API servers. These servers are hosted at the Qualys platform, also referred to as the Security Operations Center (SOC), where your account is located. If you need assistance with obtaining a Qualys account, please contact your Qualys account representative.

Users with a Qualys user account may access the API functions. When a subscription has multiple users, all users with any user role (except Contact) can use the Qualys API. Each user’s permissions correspond to their assigned user role.

Qualys user accounts that have been enabled with VIP two-factor authentication can be used with the Qualys API, however two-factor authentication will not be used when making API requests. Two-factor authentication is only supported when logging into the Qualys GUI.
Qualys API Server URL

The Qualys API URL you should use for API requests depends on the Qualys platform where your account is located.

Click here to identify your Qualys platform and get the API URL

This documentation uses the API server URL for Qualys US Platform 1 (https://qualysapi.qualys.com) in sample API requests. If you’re on another platform, please replace this URL with the appropriate server URL for your account.

Still have questions? You can easily find the API server URL for your account. Just log in to your Qualys account and go to Help > About. You’ll see this information under Security Operations Center (SOC).
Making API requests

Curl samples in our API docs
We use curl in our API documentation to show an example how to form REST API calls, and it is not meant to be an actual production example of implementation.

GET and POST Methods
Qualys API functions allow API users to submit parameters (name=value pairs) using the GET and/or POST method. There are known limits for the amount of data that can be sent using the GET method, and these limits are dependent on the toolkit used. Please refer to the individual descriptions of the API function calls to learn about the supported methods for each function.

Parameters in URLs
API parameters, as documented in this user guide, should be specified one time for each URL. In the case where the same parameter is specified multiple times in a single URL, the last parameter takes effect and the previous instances are silently ignored.

Date Format in API Results
The Qualys API has adopted a date/time format to provide consistency and interoperability of the Qualys API with third-party applications. The date format follows standards published in RFC 3339 and ISO 8601, and applies throughout the Qualys API. The date format is:

```
yyyy-mm-ddThh-mm-ssZ
```

This represents a UTC value (GMT time zone).

URL Encoding in API Code
You must URL encode variables when using the Qualys API. This is standard practice for HTTP communications. If your application passes special characters, like the single quote (‘), parentheses, and symbols, they must be URL encoded.

For example, the pound (#) character cannot be used as an input parameter in URLs. If “#” is specified, the Qualys API returns an error. To specify the “#” character in a URL you must enter the encoded value “%23”. The “#” character is considered by browsers and other Internet tools as a separator between the URL and the results page, so whatever follows an un-encoded “#” character is not passed to the Qualys API server and returns an error.

UTF-8 Encoding
The Qualys API uses UTF-8 encoding. The encoding is specified in the XML output header as shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
```
URL Elements are Case Sensitive

URL elements are case sensitive. The sample URL below will retrieve a previously saved scan report that has the reference code “scan/987659876.19876”. The parameter name “ref” is defined in lower-case characters. This URL will return the specified scan report:

ref=scan/987659876.19876

The sample URL below is incorrect and will not return the specified scan report because the parameter name “Ref” appears in mixed-case characters:

Ref=scan/987659876.19876

Decoding XML Reports

There are a number of ways to parse an XML file. Select the method which is most appropriate for your application and its users. Qualys publishes DTDs for each report on its Web site. For example, the scan list output DTD is found at the URL shown:

https://qualysapi.qualys.com/api/2.0/fo/scan/scan_list_output.dtd

The URLs to current report DTDs are included with the function descriptions in this document.

Occasionally Qualys updates the report DTDs. It is recommended that you request the most recent DTDs from the Qualys platform to decode your reports. The URLs to the report DTDs are included in this user guide.

Detailed information about each XML report is provided in the document Qualys API for VM and Compliance XML/DTD Reference

Some parts of the XML report may contain HTML tags or other special characters (such as accented letters). Therefore, many elements contain CDATA sections, which allow HTML tags to be included in the report. “High” ASCII and other non-printable characters are escaped using question marks.

API Limits

Qualys Cloud Platform enforces limits on the API calls subscription users can make. The limits apply to the use of all APIs, except “session” API (session login/logout).

API controls are applied per subscription based on your subscription’s service level. Default settings are provided and these may be customized per subscription by Qualys Support.

There’s 2 controls defined per subscription:

- Concurrency Limit per Subscription (per API). The maximum number of API calls allowed within the subscription during the configured rate limit period (as per service level).
- Rate Limit per Subscription (per API). The period of time that defines a window when API calls are counted within the subscription for each API. The window starts from the moment each API call is received by the service and extends backwards 1 hour or 1 day. Individual rate and count settings are applied (as per service level).

Click here to learn more about the controls and settings per service level.

How it works - Qualys checks the concurrency limit and rate limit each time an API request is received. In a case where an API call is received and our service determines a limit has been exceeded, the API call is blocked and an error is returned (the concurrency limit error takes precedence).

**Tracking API usage by user**

You can track API usage per user without the need to provide user credentials such as the username and password. Contact Qualys Support to get the X-Powered-By HTTP header enabled. Once enabled, the X-Powered-By HTTP header is returned for each API request made by a user. The X-Powered-By value includes a unique ID generated for each subscription and a unique ID generated for each user. See sample headers below.

Click here to learn more.

**HTTP Response Headers**

Your subscription’s API usage and quota information is exposed in the HTTP response headers generated by Qualys APIs (all APIs except “session” API).

The HTTP response headers generated by Qualys APIs are described below.

The HTTP status code “OK” (example: “HTTP/1.1 200 OK”) is returned in the header for normal (not blocked) API calls. The HTTP status code “Conflict” (example: “HTTP/1.1 409 Conflict”) is returned for API calls that were blocked.

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-RateLimit-Limit</td>
<td>Maximum number of API calls allowed in any given time period of &lt;number-seconds&gt; seconds, where &lt;number-seconds&gt; is the value of X-RateLimit-Window-Sec.</td>
</tr>
<tr>
<td>X-RateLimit-Window-Sec</td>
<td>Time period (in seconds) during which up to &lt;number-limit&gt; API calls are allowed, where &lt;number-limit&gt; is the value of X-RateLimit-Limit.</td>
</tr>
<tr>
<td>X-RateLimit-Remaining</td>
<td>Number of API calls you can make right now before reaching the rate limit &lt;number-limit&gt; in the last &lt;number-seconds&gt; seconds.</td>
</tr>
<tr>
<td>X-RateLimit-ToWait-Sec</td>
<td>The wait period (in seconds) before you can make the next API call without being blocked by the rate limiting rule.</td>
</tr>
<tr>
<td>X-Concurrency-Limit-Limit</td>
<td>Number of API calls you are allowed to run concurrently.</td>
</tr>
<tr>
<td>Header</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>X-Concurrency-Limit-Running</td>
<td>Number of API calls that are running right now (including the one identified in the current HTTP response header).</td>
</tr>
<tr>
<td>X-Powered-By</td>
<td>This header is only returned when the X-Powered-By header is enabled for your subscription. It includes a unique ID generated for each subscription and a unique ID generated for each user. Click here to learn more.</td>
</tr>
</tbody>
</table>

Sample HTTP Response Headers

Sample 1: Normal API call (API call not blocked)
Returned from API call using HTTP authentication.

```
HTTP/1.1 200 OK
Date: Fri, 22 Apr 2018 00:13:18 GMT
Server: qweb
X-RateLimit-Limit: 15
X-RateLimit-Window-Sec: 360
X-Concurrency-Limit-Limit: 3
X-Concurrency-Limit-Running: 1
X-RateLimit-ToWait-Sec: 0
X-RateLimit-Remaining: 4
Transfer-Encoding: chunked
Content-Type: application/xml
```

Sample 2: API Call Blocked (Rate Limit exceeded)
Returned from API call using HTTP authentication.

```
HTTP/1.1 409 Conflict
Date: Fri, 22 Apr 2018 00:13:18 GMT
Server: qweb
X-RateLimit-Limit: 15
X-RateLimit-Window-Sec: 360
X-Concurrency-Limit-Limit: 3
X-Concurrency-Limit-Running: 1
X-RateLimit-ToWait-Sec: 181
X-RateLimit-Remaining: 0
Transfer-Encoding: chunked
Content-Type: application/xml
```

Sample 3: API Call Blocked (Concurrency Limit exceeded)
Returned from API call using API session authentication.
HTTP/1.1 409 Conflict
Date: Fri, 22 Apr 2018 00:13:18 GMT
Server: qweb
Expires: Mon, 24 Oct 1970 07:30:00 GMT
Cache-Control: post-check=0,pre-check=0
Pragma: no-cache
X-RateLimit-Limit: 15
X-RateLimit-Window-Sec: 360
X-Concurrency-Limit-Limit: 3
X-Concurrency-Limit-Running: 3
Transfer-Encoding: chunked
Content-Type: application/xml

In case where the concurrency limit has been reached, no information about rate limits will appear in the HTTP headers.

**Sample 4: Tracking API usage through the X-Powered-By HTTP header**

HTTP/1.1 200 OK
Date: Fri, 22 Apr 2018 00:13:18 GMT
Server: qweb
X-Powered-By: Qualys:USPOD1:d9a7e94c-0a9d-c745-82e9-980877cc5043:f178af1e-4049-7fca-75584feb8e93
X-RateLimit-Limit: 15
X-RateLimit-Window-Sec: 360
X-Concurrency-Limit-Limit: 3
X-Concurrency-Limit-Running: 1
X-RateLimit-ToWait-Sec: 0
X-RateLimit-Remaining: 4
Transfer-Encoding: chunked
Content-Type: application/xml

Once X-Powered-By HTTP header is enabled, information is returned in the following format:

X-Powered-By: Qualys:<POD_ID>:<SUB_UUID>:<USER_UUID>

Where,

POD_ID is the shared POD or a PCP. Shared POD is USPOD1, USPOD2, etc.
SUB_UUID is the unique ID generated for the subscription
USER_UUID is the unique ID generated for the user

For example,

X-Powered-By: Qualys:USPOD1:d9a7e94c-0a9d-c745-82e9-980877cc5043:f178af1e-4049-7fca-75584feb8e93
You can use the USER_UUID to track API usage per user.
Activity Log

You can view the Activity Log using the Qualys user interface and the Activity Log API (/api/2.0/fo/activity_log). The Activity Log shows details about user actions taken.

To view the Activity Log, log into your Qualys account. Go to Users and click the Activity Log tab. Select Filters > Recent API Calls. You’ll see the API Processes list showing the API calls subject to the API limits (all APIs except “session” API) made by subscription users and/or updated by the service in the past week.

Tip - You can search the processes list to find API processes. You can search by process state (Queued, Running, Expired, Finished and/or Blocked), by submitted date and by last updated date. You can search for API processes that were blocked due to exceeding the API rate limit and/or the API concurrency limit.
Authentication to your account

Authentication with valid Qualys account credentials is required for making Qualys API requests to the Qualys API servers. When calling the V2 APIs (i.e. APIs with /2.0/ as URL element), users have the option to choose between session based authentication (using login and logout operations) and basic HTTP authentication (method supported for V1 APIs (i.e. APIs with /msp/ as URL element).

What do I need to know?

Using the API Session Resource
Session Login
Session Logout

What do I need to know?

Here’s some things to know about making authenticated API requests to Qualys API servers.

Required Header Parameter
The following header parameter must be included in all API calls using basic HTTP authentication and session based authentication:

"X-Requested-With: <user description, like a user agent>"

Specifying the required “X-Requested-With” parameter helps to protect Qualys API users from cross-site request forgery (CSRF) attacks.

Using Basic HTTP Authentication

Using this method, Qualys account credentials are transmitted using the “Basic Authentication Scheme” over HTTPS for each API call. For information, see the “Basic Authentication Scheme” section of RFC #2617:

http://www.faqs.org/rfcs/rfc2617.html

The exact method of implementing authentication will vary according to which programming language is used.

A sample asset/host API request (Curl) using basic HTTP authentication:

```bash
curl -H "X-Requested-With: Curl Sample" -u "acme_ab12:passwd" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/?action=list"
```
Using Session Based Authentication

Using this method, the user makes a sequence of API requests as follows (supported for V2 API calls):

Step 1: Make session login request

Use the Qualys API session resource to make a login request. Upon success, the request returns a session ID in the Set-Cookie HTTP header:

```bash
curl -H "X-Requested-With: Curl Sample" -D headers
-d "action=login&username=acme_ab12&password=PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/session/
```

Step 2: Make resource requests

Use the API resources to make API requests, as described in this user guide, and include the session ID in the cookie header for each request.

You’ll notice the session cookie (QualysSession) was extracted from the “headers” file contents returned from the session login API call (Step 1 above):

```bash
curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=71e6cda2a35d2cd404cdaf305ea0208; path=/api;
secure" -d "action=list"
"https://qualysapi.qualys.com/api/2.0/fo/report/
```

Step 3: Make session logout request

Once logged in to Qualys you can make multiple API requests. Use the Qualys API session resource to logout of the current session. Logging out of the session closes the open session and ensures secure, ongoing access to your account. Access may be denied if a user makes too many session login requests without closing sessions properly:

```bash
curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=10b8eb6d4553b4d1ecb860c2b3c247d4; path=/api;
secure" -d "action=logout"
"https://qualysapi.qualys.com/api/2.0/fo/session/
```

Using the API Session Resource

Sessions created using the Qualys API via the session resource are equivalent in every way to sessions created by users logging into the Qualys user interface. Too many open sessions, whether created via the API and/or via user interface login, will lock out new session login attempts from both interfaces (user and API).
The request URL has several elements. The following elements appear in every request URL based on the API V2 architecture.

<table>
<thead>
<tr>
<th>URL element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qualysapi.qualys.com:443</td>
<td>FQDN of the Qualys API server and option port (443 if specified).</td>
</tr>
<tr>
<td>api</td>
<td>Qualys Application component name.</td>
</tr>
<tr>
<td>2.0</td>
<td>Qualys API version number.</td>
</tr>
<tr>
<td>fo</td>
<td>Qualys interface component name.</td>
</tr>
<tr>
<td>session</td>
<td>scan</td>
</tr>
<tr>
<td>action={value}</td>
<td>Qualys API resource-specific action. In the sample session login URL above, the action is “login”.</td>
</tr>
</tbody>
</table>

**Session Login Request**

The session login request includes the Qualys user login credentials, the request URL, and the location where the HTTP response headers will be saved.

The sample API call below saves the HTTP headers in a local file named “headers”:

```bash
curl -H "X-Requested-With: Curl Sample" -D headers -d "action=login&username=acme_ab12&password=<PASSWORD>" "https://qualysapi.qualys.com/api/2.0/fo/session/
```

If you do not wish to store this information in the "headers" file, you can save the HTTP header in a cookie as shown below:

```bash
curl -H "X-Requested-With: Curl Sample" -c cookie.txt -d "action=login&username=acme_ab12&password=<PASSWORD>" "https://qualysapi.qualys.com/api/2.0/fo/session/
```

Upon success, the sample Qualys API call returns an XML response with the message “Logged in” and the Qualys API session ID in the Set-Cookie HTTP header. See “HTTP Response Headers” for further information.

**Resource Requests**

When session based authentication is used, the session cookie returned in the XML response from the session login request must be included in the cookie header of subsequent API requests. Multiple API requests can be made using the same session cookie (this is supported using V2 API requests).

The resource request includes the Qualys user login credentials, the Qualys API session ID, the request URL, and the location where the HTTP response headers are saved.

The sample API request below is used to request a list of reports in the user’s Report Share storage space. You’ll notice the session cookie (QualysSession) was extracted from the “headers” file contents returned from the session login API call.
Authentication to your account
Using Session Based Authentication

curl -H "X-Requested-With: Curl Sample"
   -d "action=list"
   -b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/report/"

If you saved the HTTP response headers (from the session login request) in a cookie file, make an API request to obtain the cookie from the cookie file as shown below:

curl -H "X-Requested-With: Curl Sample"
   -d "action=list"
   -b "cookie.txt" "https://qualysapi.qualys.com/api/2.0/fo/report/"

Upon success, the sample report list API call returns an XML response listing the reports in the user’s Report Share. In progress and completed reports are included.

HTTP Response Headers
These API requests return HTTP response headers: session login requests, session logout requests, and fetch (download) report requests. These requests provide information to the third party application about the XML output.

Sample XML output showing HTML response headers returned from a session logout request:

HTTP/1.1 200 OK
Date: Wed, 20 Jun 2007 16:21:03 GMT
Server: qweb/3.3h
Set-Cookie: QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure
Expires: Mon, 24 Oct 1970 07:30:00 GMT
Cache-Control: post-check=0,pre-check=0
Pragma: no-cache
Connection: close
Transfer-Encoding: chunked
Content-Type: text/xml

Sample XML output showing HTML response headers returned from a fetch (download) report request, where the report format is HTML:

HTTP/1.1 200 OK
Date: Wed, 20 Jun 2007 16:36:42 GMT
Server: qweb/3.3h
Expires: Mon, 24 Oct 1970 07:30:00 GMT
Cache-Control: post-check=0,pre-check=0
Pragma: no-cache
Content-Disposition: attachment;
filename=scan_report__1182357402.zip
Content-length: 98280
Connection: close
Expires HTTP Header - For the Expires header, Qualys complies with RFC #2109 and sets the Expires date to an old date (a date long in the past). Currently Qualys sets the Expires date to “Mon, 24 Oct 1970 07:30:00 GMT”. Note that Qualys cookie expiration is managed on the server side, and Qualys does not rely on clients to drop their expired cookies.

**Session Logout Request**

A sample session logout request (POST method) is shown below. Upon success, the sample Qualys API call returns an XML response with the message "Logged out".

```bash
curl -H "X-Requested-With: Curl Sample"
   -d "action=logout"
   -b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
   secure" "https://qualysapi.qualys.com/api/2.0/fo/session/"
```

See “Session Logout” below for further information.

**Session Timeout**

Every Qualys user account has a session timeout setting. This setting is configurable at the subscription level by Manager users in the Qualys user interface (go to Users > Setup > Security). For a new subscription, this is set to 60 minutes.

The session timeout applies to sessions started using the user interface and sessions started using the Qualys APIs, including APIs based on the new API architecture.

When you launch a scan or report (using Report Share), the task is launched in the background, and processing does not timeout until the task has completed.

**Session Login**

/api/2.0/fo/session/?action=login

[POST]

Make a request to Qualys API server for session login.

A session login request is used to authenticate to the Qualys API and receive a Qualys API session ID, which must be included in the cookie header of subsequent API resource requests.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=login</td>
<td>(Required) A flag used to make a session login request.</td>
</tr>
<tr>
<td>username</td>
<td>(Required) The user name (login) of a Qualys user account.</td>
</tr>
</tbody>
</table>
A sample session login request (POST method) is shown below. Upon success, the sample Qualys API call returns an XML response with the message “Logged in” and the Qualys API session ID as shown.

```bash
curl -H "X-Requested-With: Curl Sample" -D headers.4 -d "action=login&username=acme_ab12&password=<PASSWORD>" "https://qualysapi.qualys.com/api/2.0/fo/session/

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE GENERIC SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">

<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2007-06-20T16:21:04Z</DATETIME>
    <TEXT>Logged in</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>

```
cat headers.4

HTTP/1.1 200 OK
Date: Wed, 20 Jun 2007 16:21:03 GMT
Server: qweb/3.3h
Set-Cookie: QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure
Expires: Mon, 24 Oct 1970 07:30:00 GMT
Cache-Control: post-check=0,pre-check=0
Pragma: no-cache

---

**Parameter** | **Description**
--- | ---
password | (Required) The password of a Qualys user account.

When using `-d` in the curl request for login, you must URL encode any special characters in the password. For example, if your password is `Peas+Carrots` then you must specify it as `password=Peas%2BCarrots` or authentication will not be successful.

When using `-u` in the curl request for login, you can enter the password as is without URL encoding special characters. Using the same example, you’d specify `password=Peas+Carrots` as part of the request.

**echo_request=0|1** | (Optional) Specifies whether to echo the request’s input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
Connection: close
Transfer-Encoding: chunked
Content-Type: text/xml

Session Logout

/api/2.0/fo/session/?action=logout

[POST]

Make a request to Qualys API server for session logout.

When you’re done making V2 API resource requests, the third party application must make a session logout request. This results in closing the session ID for the user’s account, preventing future API requests from running.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=logout</td>
<td>(Required) A flag used to make a session logout request.</td>
</tr>
<tr>
<td>echo_request=[0</td>
<td>1]</td>
</tr>
</tbody>
</table>

A sample session logout request (POST method) is shown below. Upon success, the sample Qualys API call returns an XML response with the message “Logged out” as shown.

curl -H "X-Requested-With: Curl Sample"
-d "action=logout"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/session/"

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE GENERIC SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2007-06-20T21:50:37Z</DATETIME>
    <TEXT>Logged out</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>

cat headers.18)

HTTP/1.1 200 OK
Date: Wed, 20 Jun 2007 21:50:36 GMT
Server: qweb/3.3h
Expires: Mon, 24 Oct 1970 07:30:00 GMT
Cache-Control: post-check=0,pre-check=0
Pragma: no-cache
Set-Cookie: QualysSession=71e6cda2a35d2cd404cddaf305ea0208;
expires=Wed, 13-Jun-2007 21:50:37 GMT; path=/fo
Connection: close
Transfer-Encoding: chunked
Content-Type: text/xml
Scans

Launch and manage vulnerability scans, compliance scans, discovery scans (maps).

VM Scans | Compliance Scans | Cloud Perimeter Scans
VM Scan Schedules | PC Scan Schedules
Scan List Parameters | Scan Parameters | Cloud Perimeter Scan Parameters | Scan Schedule Parameters
VM Scan Statistics
VM Scan Summary | Scan Summary
Scanner Details
Share PCI Scan
Discovery Scans (maps) | Domain List | Add/Edit Domain
VM Scans

The VM Scan API (/api/2.0/fo/scan/) is used to obtain a list of vulnerability scans in your account and to take actions on them like cancel, pause, resume, and fetch (download) finished results.

Express Lite: This API is available to Express Lite users.

Permissions

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>Manage scans on all IPs in the subscription.</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>Launch, list and fetch scans on IPs in the user’s business unit. And take actions on scans launched by users in the same business unit (cancel, pause, resume and delete).</td>
</tr>
<tr>
<td>Scanner</td>
<td>Launch, list and fetch scans on IPs in the user’s account. And take actions on scans that the user owns (cancel, pause, resume and delete).</td>
</tr>
<tr>
<td>Reader</td>
<td>View scans with targets containing IPs in the user’s account. Download scan results when the target includes at least one IP in the user’s account.</td>
</tr>
<tr>
<td>Auditor</td>
<td>No permissions.</td>
</tr>
</tbody>
</table>

VM Scan List

/api/2.0/fo/scan/?action=list

[GET] [POST]

List vulnerability scans in the user’s account. By default the XML output lists scans launched in the past 30 days.

Input Parameters

The input parameters for requesting a VM scan list are shown below. See Scan List Parameters for complete details.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=list (required), echo_request</td>
</tr>
<tr>
<td>Scan List Filters</td>
<td>scan_ref, state, processed, type, target, user_login,</td>
</tr>
<tr>
<td></td>
<td>launched_after_datetime, launched_before_datetime,</td>
</tr>
<tr>
<td></td>
<td>scan_type=certview, scan_type=ec2certview, client_id and</td>
</tr>
<tr>
<td></td>
<td>client_name (only for Consultant type subscriptions)</td>
</tr>
<tr>
<td>Show/Hide Information</td>
<td>show_ags, show_op, show_status, show_last, ignore_target</td>
</tr>
</tbody>
</table>
Samples
List all scans in the user account.

curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/
?action=list&echo_request=1&show_ags=1&show_op=1"

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCAN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/scan_list_output.dtd">
<SCAN_LIST_OUTPUT>
<REQUEST>
<USER_LOGIN>acme_ab</USER_LOGIN>
<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/scan/</RESOURCE>
<PARAM_LIST>
<PARAM>
<KEY>action</KEY>
<VALUE>list</VALUE>
</PARAM>
<PARAM>
<KEY>echo_request</KEY>
<VALUE>1</VALUE>
</PARAM>
<PARAM>
<KEY>show_ags</KEY>
<VALUE>1</VALUE>
</PARAM>
<PARAM>
<KEY>show_op</KEY>
<VALUE>1</VALUE>
</PARAM>
</PARAM_LIST>
</REQUEST>
<RESPONSE>
<SCAN_LIST>
<SCAN>
<REF>scan/1187117392.587</REF>
<TYPE>On-Demand</TYPE>
<TITLE><![CDATA[Web Servers 09/25]]></TITLE>
<USER_LOGIN>acme_ab</USER_LOGIN>
<LAUNCH_DATETIME>2018-05-25T08:10:43Z</LAUNCH_DATETIME>
List all running scans that were launched by the user with the login ID "acme_ab":

curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=71e6cda2a35d2cd404cdaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/
?action=list&state=Running&user_login=acme_ab"

List all scheduled scans that were launched after June 5, 2018.

curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=71e6cda2a35d2cd404cdaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/
?action=list&type=Scheduled&launched_after_datetime=2018-06-05"
List all scans for AFCO Company client (only for Consultant type subscriptions).

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" "https://qualysapi.qualys.com/api/2.0/fo/scan/?action=list&client_name=AFCO Company"
```

**DTD**

```xml
<platform API server>/api/2.0/fo/scan/scan_list_output.dtd
```

**List Last ‘N’ Scan References for a Schedule**

`api/2.0/fo/scan/schedules/runhistory/?action=list`

[GET]

Provides a list of the most recent "N" scan references associated with a particular schedule ID, allowing you to monitor and track automated scans initiated for that schedule.

Note: This API does not work for MAP schedule scan ids.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>schedule_scan_ids</td>
<td>(Required) Specify schedule ids in a comma-separated list. Maximum 500 schedule ids are supported.</td>
</tr>
<tr>
<td>output_format</td>
<td>(Optional) Specify output file format. Default value is &quot;xml&quot; format. Supported values are &quot;xml&quot; &amp; &quot;json&quot;.</td>
</tr>
<tr>
<td>schedule_executions_count</td>
<td>(Optional) Specify a schedule execution count. You can specify a value form 1 to 10. When not specified, 3 is taken as the default.</td>
</tr>
</tbody>
</table>

**Sample**

List the last ‘n’ scan references for the given schedule IDs.

**API request:**

```bash
curl --location \
'https://qualysapi.qualys.com/api/2.0/fo/scan/schedules/runhistory/?action=list&output_format=xml&schedule_executions_count=4&schedule_scan_ids=99446%2C85403%2C144180%2C144181%2C6666%2C657657%2C125485' \
--header 'X-Requested-With: test' \
--header 'Authorization: Basic <token>'
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8"?>
```
<!DOCTYPE SCHEDULES_RUN_HISTORY SYSTEM "<platform_API_server>/api/2.0/fo/scan/schedules/runhistory/output.dtd">
<SCHEDULES_RUN_HISTORY>
  <RESPONSE>
    <DATETIME>2023-01-09T14:57:51Z</DATETIME>
    <SCHEDULE_LIST>
      <SCHEDULE id="85403">
        <SCHEDULE_ID>85403</SCHEDULE_ID>
        <SCHEDULE_RUN_HISTORY>
          <SCHEDULE_RUN>
            <SCAN_LAUNCH_INFO>
              <SCAN_ID>99408</SCAN_ID>
              <SCAN_REFERENCE>scan/1634824016.99408</SCAN_REFERENCE>
              <LAUNCH_DATETIME>2021-10-21 13:46:56</LAUNCH_DATETIME>
              <TITLE>test9090</TITLE>
              <TARGET>10.20.30.40</TARGET>
              <STATUS>FINISHED</STATUS>
              <DURATION>86 Seconds</DURATION>
              <NBHOST>0</NBHOST>
              <SUBSCRIPTION_ID>237630</SUBSCRIPTION_ID>
              <OPTION_PROFILE_TITLE>Initial Options</OPTION_PROFILE_TITLE>
              <IS_PROCESSED>1</IS_PROCESSED>
            </SCAN_LAUNCH_INFO>
          </SCHEDULE_RUN>
        </SCHEDULE_RUN_HISTORY>
      </SCHEDULE>
      <SCHEDULE id="99446">
        <SCHEDULE_ID>99446</SCHEDULE_ID>
        <SCHEDULE_RUN_HISTORY>
          <SCHEDULE_RUN>
            <SCAN_LAUNCH_INFO>
              <SCAN_ID>99561</SCAN_ID>
              <SCAN_REFERENCE>scan/1635442781.99561</SCAN_REFERENCE>
              <LAUNCH_DATETIME>2021-10-28 17:39:41</LAUNCH_DATETIME>
              <TITLE>test_555</TITLE>
              <TARGET>list of target ips</TARGET>
              <STATUS>FINISHED</STATUS>
              <DURATION>434 Seconds</DURATION>
              <NBHOST>0</NBHOST>
              <SUBSCRIPTION_ID>237630</SUBSCRIPTION_ID>
            </SCAN_LAUNCH_INFO>
          </SCHEDULE_RUN>
        </SCHEDULE_RUN_HISTORY>
      </SCHEDULE>
    </SCHEDULE_LIST>
  </RESPONSE>
</SCHEDULES_RUN_HISTORY>
<OPTION_PROFILE_TITLE>Initial Options</OPTION_PROFILE_TITLE>
  <IS_PROCESSED>1</IS_PROCESSED>
</SCAN_LAUNCH_INFO>
</SCHEDULE_RUN>
<SCAN_LAUNCH_INFO>
  <SCAN_ID>99553</SCAN_ID>
  <SCAN_REFERENCE>scan/1635431408.99553</SCAN_REFERENCE>
  <LAUNCH_DATETIME>2021-10-28 14:30:08</LAUNCH_DATETIME>
  <TITLE>test9090</TITLE>
  <TARGET>10.20.30.40</TARGET>
  <STATUS>FINISHED</STATUS>
  <DURATION>79 Seconds</DURATION>
  <NBHOST>0</NBHOST>
  <SUBSCRIPTION_ID>237630</SUBSCRIPTION_ID>
  <OPTION_PROFILE_TITLE>Initial Options</OPTION_PROFILE_TITLE>
  <IS_PROCESSED>1</IS_PROCESSED>
</SCAN_LAUNCH_INFO>
</SCHEDULE_RUN>
</SCHEDULE_RUN_HISTORY>
</SCHEDULEList>
</SCHEDULEs_RUN_HISTORY>
Launch VM Scan

/api/2.0/fo/scan/?action=launch

[POST]

Launch vulnerability scan in the user’s account.

Good to Know
- The Launch Scan API is asynchronous. When you make a request to launch a scan using this API, the service will return a scan reference ID right away and the call will quit without waiting for the complete scan results.
- When you launch a VM scan using the API, we check to see if the IPs in the scan target are available to the user making the scan request. To determine this, we check that each IP is in the subscription, in the VM license, and in the user’s assigned scope. If any IP in the target is not available to the user, then it will be skipped from the scan job.

For example, let’s say you specify the IP range 10.10.10.100-10.10.10.120, but IPs 10.10.10.115 and 10.10.10.120 are not available to you. In this case, we will launch the scan on 10.10.10.100-10.10.10.114, 10.10.10.116-10.10.10.119, and we’ll skip 10.10.10.115 and 10.10.10.120.
- Using networks? Choose the Global Default Network to scan IPs on your network perimeter.

Input Parameters
The input parameters for launching a VM scan are shown below. See Scan Parameters for complete details.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=launch (required), echo_request, runtime_http_header</td>
</tr>
<tr>
<td>Scan Title</td>
<td>scan_title</td>
</tr>
<tr>
<td>Option Profile</td>
<td>option_id or option_title</td>
</tr>
<tr>
<td>Scanner Appliance</td>
<td>isscanner_id or isscanner_name, ec2_instance_ids</td>
</tr>
<tr>
<td>Processing Priority</td>
<td>priority</td>
</tr>
<tr>
<td>Asset IPs/Groups</td>
<td>ip, asset_group_ids, asset_groups, exclude_ip_per_scan, default_scanner, scanners_in_ag</td>
</tr>
</tbody>
</table>
### Sample - Launch scan on IP address

**API request:**

```bash
curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWORD" -X "POST"
-d "action=launch&scan_title=My+Vulnerability+Scan&ip=10.10.10.10&option_id=43165&iscanner_name=scanner1"
"https://qualysapi.qualys.com/api/2.0/fo/scan/" > outputfile.txt
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-01-15T21:32:40Z</DATETIME>
    <TEXT>New vm scan launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>136992</VALUE>
      </ITEM>
      <ITEM>
        <KEY>REFERENCE</KEY>
        <VALUE>scan/1358285558.36992</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Launch Scan Samples

API request (FQDN only):

```bash
curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWD" -X "POST" -d "action=launch&option_title=Initial+Options&fqdn=domain.qualys.com &iscanner_name=scanner_us" "https://.qualysapi.qualys.com/api/2.0/fo/scan/" > outputfile.txt
```

API request (FQDN and asset group):

```bash
curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWD" -X "POST" -d "action=launch&option_title=Initial+Options&fqdn=domain.qualys.com &iscanner_name=scanner_us&scan_title=My+Scan&asset_groups=My+AG" "https://qualysapi.qualys.com/api/2.0/fo/scan/" > outputfile.txt
```

Sample - Launch scan using asset tags

API request:

```bash
curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWD" -X "POST" -d "action=launch&scan_title=My+Vulnerability+Scan&target_from=tags&tag_set_by=name&tag_set_include=Windows&option_id=43165&iscanner_name=scanner1" "https://qualysapi.qualys.com/api/2.0/fo/scan/" > file.txt
```

Sample - Launch scan using All Scanners in Network

API request:

```bash
curl -u "username:password" -H "X-Requested-With:curl demo" -d "action=launch&scan_title=scan3&option_title=Initial+Options&ip_network_id=12807913&scanners_in_network=I&asset_groups=AG1-GDN" "https://qualysapi.qualys.com/api/2.0/fo/scan/"
```

Launch VM Scan on EC2 assets

`/api/2.0/fo/scan/?action=launch`

[POST]

Launch vulnerability scan on your Amazon EC2 hosts (in your Amazon Web Services account).

A few things to consider...

- EC2 Scanning must be enabled for your Qualys account.
- Managers and Unit Managers can launch EC2 scans.
- Before scanning you’ll need to complete some set up steps. See [Securing Amazon Web Services with Qualys](#)
### Input Parameters

The input parameters for launching an EC2 scan are shown below. See Scan Parameters for complete details.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=launch (required), echo_request</td>
</tr>
<tr>
<td>Scan Title</td>
<td>scan_title</td>
</tr>
<tr>
<td>EC2 environment</td>
<td>connector_name (required), ec2_endpoint (required)</td>
</tr>
<tr>
<td>Option Profile</td>
<td>option_id or option_title</td>
</tr>
<tr>
<td>Scanner Appliance</td>
<td>iscanner_id or iscanner_name</td>
</tr>
<tr>
<td>Processing Priority</td>
<td>priority</td>
</tr>
<tr>
<td>Target Hosts</td>
<td>target_from=tags</td>
</tr>
<tr>
<td></td>
<td>Use tags to select the EC2 hosts you want to scan.</td>
</tr>
</tbody>
</table>

Note: You can use either ec2_instance_ids or tags parameter or both.

These parameters provide separate options for including and excluding tags for network IP ranges.

- `use_ip_nt_range_tags_include={0|1}` (default in bold)
- `use_ip_nt_range_tags_exclude={0|1}` (default in bold)

Important - This cannot be set to "1" for EC2 scanning.

This parameter has been replaced with the include/exclude options above but it is still supported.

- `use_ip_nt_range_tags={0|1}` (default in bold)

Important - This cannot be set to "1" for EC2 scanning.

These tag parameters are used to select tags:

- `tag_set_include={tag1,tag2,...}` (required)
- `tag_set_exclude={tag1,tag2,...}` (optional)
- `tag_include_selector={any|all}` (default in bold)
- `tag_exclude_selector={any|all}` (default in bold)
- `tag_set_by={id|name}` (default in bold)

- `ec2_instance_ids={value}`

The ID of the target EC2 instance to launch the VM or compliance scan. Multiple ec2 instance ids are comma separated. You can add up to maximum 10 instance Ids.

---

### Sample - Launch EC2 Vulnerability scan

Launch an EC2 vulnerability scan using the connector “EC2_Connector” on assets that match tags with IDs 1558997 and 1559222.

**API request:**

```
curl -X "X-Requested-With: Curl" -u "USERNAME:PASSWD" -X "POST" -d "action=launch&scan_title=My+EC2+Scan&connector_name=EC2_Connector&ec2_endpoint=us-east-1&target_from=tags&use_ip_nt_range_tags=0&tag_include_selector=any&tag_set_by=id&tag_set_include=1558997,1559222&option_id=43165&iscanner_name=EC2-1" "https://qualysapi.qualys.com/api/2.0/fo/scan/" > outputfile.txt
```
Sample - Launch EC2 Vulnerability scan for EC2 instance

Launch a VM scan on EC2 instances using the parameter ec2_instance_ids.

This sample is for a vulnerability scan with a mix of valid and invalid instance IDs. The scan is launched on the valid instance IDs and the invalid instance IDs are listed in the output with the reasons they were considered invalid. Some did not belong to the EC2 environment and some were not activated for VM.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -d "action=launch&scan_title=Sample2&connector_name=EC2 Connector&ec2_endpoint=us-east-1&option_title=Initial Options&iscanner_name=EC2_Scanner&ec2_instance_ids=i-01f234ce567ae890f,i-0be12cb3da4567e8a,i-0d1f23d4ba5c67e8b,i-0123e456f7890f123,i-012f3ceb4a5d6789d,i-0c123e4f567890123,i-012345678bba89012,i-01ba23a45cba678af,i-012345678dfe90efe,i-0ab12e345fbaa7eb7" "https://qualysapi.qualys.com/api/2.0/fo/scan/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/scan/dtd/launch_output.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-02-25T21:32:40Z</DATETIME>
    <TEXT>New vm scan launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>136992</VALUE>
      </ITEM>
      <ITEM>
        <KEY>REFERENCE</KEY>
        <VALUE>scan/1358285558.36992</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
<RESPONSE>
<DATETIME>2021-11-19T09:13:21Z</DATETIME>
<TEXT>New vm scan launched</TEXT>
<NOTIFICATION>The following instances were skipped because they do not belong to the selected EC2 environment: i-012f3ceb4a5d6789d, i-0c123e4f567890123, i-012345a67bba89012. The following instances were skipped because they are not activated for VM: i-01ba23a45cba678af, i-012345678dfc90efe, i-0ab12e3456baadeb7.</NOTIFICATION>
<ITEM_LIST>
<ITEM>
<KEY>ID</KEY>
<VALUE>1140800</VALUE>
</ITEM>
<ITEM>
<KEY>REFERENCE</KEY>
<VALUE>scan/1637313199.40800</VALUE>
</ITEM>
</ITEM_LIST>
</RESPONSE>

Manage VM Scans

/api/2.0/fo/scan/?action={action}

Take actions on vulnerability scans in their account, like cancel, pause, resume, delete and fetch completed scan results.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) One action required for the request: cancel - Stop a scan in progress (POST method) pause - Stop a scan in progress and change status to “Paused” (POST method) resume - Restart a scan that has been paused (POST method) delete - Delete a scan in your account (POST method) fetch - Download scan results for a scan with status of “Finished”, “Canceled”, “Paused” or “Error” (GET or POST method)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_ref={value}</td>
<td>(Required) The scan reference for a vulnerability scan. This will have the format: scan/nnnnnnnnnn.nnnnn</td>
</tr>
</tbody>
</table>
## Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) An action for the request: cancel - stop a scan in progress, ”Running” or “Paused” pause - stop a scan in progress and change status to “Paused” resume - restart a scan that has been paused fetch - download scan results for a scan with the status “Finished”, “Cancled”, “Paused” or “Error”.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_ref={value}</td>
<td>(Required) Specifies a scan reference. A scan reference has the format “scan/987659876.19876”.</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional for a fetch request) Show only certain IP addresses/ranges in the scan results. One or more IPs/ranges may be specified. A range entry is specified using a hyphen (for example, 10.10.10.1-10.10.10.20). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>mode={brief</td>
<td>extended}</td>
</tr>
<tr>
<td>output_format={csv</td>
<td>json</td>
</tr>
<tr>
<td>client_id={value}</td>
<td>(Optional for fetch request) Id assigned to the client (Consultant type subscription only). Parameter client_id or client_name may be specified for the same request.</td>
</tr>
<tr>
<td>client_name={value}</td>
<td>(Optional for fetch request) Name of the client (Consultant type subscription only). Parameter client_id or client_name may be specified for the same request.</td>
</tr>
</tbody>
</table>

## Samples - Take actions on scans

Cancel a scan (POST method) is shown below.

```bash
curl -H "X-Requested-With: Curl Sample"
-d "action=cancel&scan_ref=234234234.12345"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/
```
Pause a scan (POST method) is shown below.

curl -H "X-Requested-With: Curl Sample"
-d "action=pause&scan_ref=234234234.12345"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/"

Resume a scan (POST method) is shown below.

curl -H "X-Requested-With: Curl Sample"
-d "action=resume&scan_ref=234234234.12345"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/"

Fetch/download a scan result is shown below.

curl -H "X-Requested-With: Curl Sample"
-d "action=fetch&scan_ref=234234234.12345"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/"

**DTD**

`<platform API server>/api/2.0/simple_return.dtd`
Compliance Scans

The Compliance Scan API (/api/2.0/fo/scan/compliance/) is used to launch compliance scans, get a list of compliance scans in your account and manage them. The SCAP Scan API (/api/2.0/fo/scan/scap/) is used to get a list of SCAP scans in your account.

Permissions

Note: The Compliance Scan APIs are available as part of one of the following subscription combinations only:

- PC and API add-on
- PC, SCA, and API add-on
- VMDR, SCA, and API add-on

Role-based user permissions are described below.

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>Manage compliance scans on all compliance IPs in the subscription.</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>When the “Manage compliance” permission is enabled in the user’s account settings: 1) ability to launch, list and fetch compliance scans on IPs in the user’s business unit, 2) ability to take actions on scans launched by users in the same business unit (cancel, pause, resume and delete).</td>
</tr>
<tr>
<td>Scanner</td>
<td>When the “Manage compliance” permission is enabled in the user’s account settings: 1) ability to launch, list and fetch compliance scans on IPs in the user’s account, 2) ability to take actions on scans that the user owns (cancel, pause, resume and delete).</td>
</tr>
<tr>
<td>Reader</td>
<td>No permissions to manage compliance scans.</td>
</tr>
<tr>
<td>Auditor</td>
<td>No permissions to manage compliance scans.</td>
</tr>
</tbody>
</table>
Compliance Scan List
/api/2.0/fo/scan/compliance/ with action=list

[GET]  [POST]

List of compliance scans in your account. By default the XML output lists scans launched in the past 30 days.

The input parameters for requesting a PC scan list are below. See Scan List Parameters for complete details.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=list (required), echo_request</td>
</tr>
<tr>
<td>Scan List Filters</td>
<td>scan_id (compliance scan ID), scan_ref, state, processed, type, target, user_login, launched_after_datetime, launched_before_datetime, client_id and client_name (only for Consultant type subscriptions)</td>
</tr>
<tr>
<td>Show Information</td>
<td>show_ags, show_op, show_status, show_last</td>
</tr>
</tbody>
</table>

API Request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=list&state=Finished&scan_ref=compliance/1344842952.1340"
"https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCAN_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/scan/scan_list_output.dtd ">
<SCAN_LIST_OUTPUT>
<RESPONSE>
<DATETIME>2018-06-12T07:28:46Z</DATETIME>
<SCAN_LIST>
<SCAN>
<ID>3332486</ID>
<REF>compliance/1344842952.1340</REF>
<TYPE>Scheduled</TYPE>
<TITLE><![CDATA[MY PC Scan]]></TITLE>
<User_LOGIN>USERNAME</User_LOGIN>
<LAUNCH_DATETIME>2018-05-13T07:30:09Z</LAUNCH_DATETIME>
<DURATION>00:06:29</DURATION>
<PROCESSED>1</PROCESSED>
<STATUS>
<STATE>Finished</STATE>
</SCAN>
</SCAN_LIST>
</RESPONSE>
</SCAN_LIST_OUTPUT>
```
List Last ‘N’ Scan References for a Schedule

**api/2.0/fo/scan/schedules/runhistory/?action=list [GET]**

Provides a list of the most recent "N" scan references associated with a particular schedule ID, allowing you to monitor and track automated scans initiated for that schedule.

Note: This API does not work for MAP schedule scan ids.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>schedule_scan_ids</td>
<td>(Required) Specify schedule ids in a comma-separated list. Maximum 500 schedule ids are supported.</td>
</tr>
<tr>
<td>output_format</td>
<td>(Optional) Specify output file format. Default value is “xml” format. Supported values are “xml” &amp; “json”.</td>
</tr>
<tr>
<td>schedule_executions_count</td>
<td>(Optional) Specify a schedule execution count. You can specify a value form 1 to 10. When not specified, 3 is taken as the default.</td>
</tr>
</tbody>
</table>

**Sample**

List the last 'n' scan references for the given schedule IDs.

**API request:**

```bash
curl --location
  '<qualys_base_url>/api/2.0/fo/scan/schedules/runhistory/?action=list&output_format=xml&schedule_executions_count=4&schedule_scan_ids=99446%2C85403%2C144180%2C144181%2C6666%2C657657%2C125485' \\n  --header 'X-Requested-With: test' \\n  --header 'Authorization: Basic <token>'
```

43
API request:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SCHEDULES_RUN_HISTORY SYSTEM "<platform_API_server>/api/2.0/fo/scan/schedules/runhistory/output.dtd">

<SCHEDULES_RUN_HISTORY>
  <RESPONSE>
    <DATETIME>2023-01-09T14:57:51Z</DATETIME>
    <SCHEDULE_LIST>
      <SCHEDULE id="85403">
        <SCHEDULE_ID>85403</SCHEDULE_ID>
        <SCHEDULE_RUN_HISTORY>
          <SCHEDULE_RUN>
            <SCAN_LAUNCH_INFO>
              <SCAN_ID>99408</SCAN_ID>
              <SCAN_REFERENCE>scan/1634824016.99408</SCAN_REFERENCE>
              <LAUNCH_DATETIME>2021-10-21 13:46:56</LAUNCH_DATETIME>
              <TITLE>test9090</TITLE>
              <TARGET>10.20.30.40</TARGET>
              <STATUS>FINISHED</STATUS>
              <DURATION>86 Seconds</DURATION>
              <NBHOST>0</NBHOST>
              <SUBSCRIPTION_ID>237630</SUBSCRIPTION_ID>
              <OPTION_PROFILE_TITLE>Initial Options</OPTION_PROFILE_TITLE>
              <IS_PROCESSED>1</IS_PROCESSED>
            </SCAN_LAUNCH_INFO>
          </SCHEDULE_RUN>
        </SCHEDULE_RUN_HISTORY>
      </SCHEDULE>
      <SCHEDULE id="99446">
        <SCHEDULE_ID>99446</SCHEDULE_ID>
        <SCHEDULE_RUN_HISTORY>
          <SCHEDULE_RUN>
            <SCAN_LAUNCH_INFO>
              <SCAN_ID>99561</SCAN_ID>
              <SCAN_REFERENCE>scan/1635442781.99561</SCAN_REFERENCE>
              <LAUNCH_DATETIME>2021-10-28 17:39:41</LAUNCH_DATETIME>
              <TITLE>test_555</TITLE>
              <TARGET>list of target ips</TARGET>
              <STATUS>FINISHED</STATUS>
            </SCAN_LAUNCH_INFO>
          </SCHEDULE_RUN>
        </SCHEDULE_RUN_HISTORY>
      </SCHEDULE>
    </SCHEDULE_LIST>
  </RESPONSE>
</SCHEDULES_RUN_HISTORY>
```
<DURATION>434 Seconds</DURATION>  
<NBHOST>0</NBHOST>  
<SUBSCRIPTION_ID>237630</SUBSCRIPTION_ID>  
<Option_PROFILE_TITLE>Initial Options</Option_PROFILE_TITLE>  
<IS_PROCESSED>1</IS_PROCESSED>  
</SCAN_LAUNCH_INFO>  
</SCHEDULE_RUN>  
<SCAN_LAUNCH_INFO>  
<SCAN_ID>99554</SCAN_ID>  
<SCAN_REFERENCE>scan/1635431408.99554</SCAN_REFERENCE>  
<LAUNCH_DATETIME>2021-10-28 14:30:08</LAUNCH_DATETIME>  
<TITLE>test_898</TITLE>  
<TARGET>10.20.30.40</TARGET>  
<STATUS>FINISHED</STATUS>  
<DURATION>9 Seconds</DURATION>  
<NBHOST>0</NBHOST>  
<SUBSCRIPTION_ID>237630</SUBSCRIPTION_ID>  
<Option_PROFILE_TITLE>Initial Options</Option_PROFILE_TITLE>  
<IS_PROCESSED>1</IS_PROCESSED>  
</SCAN_LAUNCH_INFO>  
</SCHEDULE_RUN>  
<SCAN_LAUNCH_INFO>  
<SCAN_ID>99553</SCAN_ID>  
<SCAN_REFERENCE>scan/1635431408.99553</SCAN_REFERENCE>  
<LAUNCH_DATETIME>2021-10-28 14:30:08</LAUNCH_DATETIME>  
<TITLE>test9090</TITLE>  
<TARGET>10.20.30.40</TARGET>  
<STATUS>FINISHED</STATUS>  
<DURATION>79 Seconds</DURATION>  
<NBHOST>0</NBHOST>  
<SUBSCRIPTION_ID>237630</SUBSCRIPTION_ID>  
<Option_PROFILE_TITLE>Initial Options</Option_PROFILE_TITLE>  
<IS_PROCESSED>1</IS_PROCESSED>  
</SCAN_LAUNCH_INFO>  
</SCHEDULE_RUN>  
<SCAN_LAUNCH_INFO>  
<SCAN_ID>99554</SCAN_ID>  
<SCAN_REFERENCE>scan/1635431408.99554</SCAN_REFERENCE>  
<LAUNCH_DATETIME>2021-10-28 14:30:08</LAUNCH_DATETIME>  
<TITLE>test898</TITLE>  
<TARGET>10.20.30.40</TARGET>  
<STATUS>FINISHED</STATUS>  
<DURATION>9 Seconds</DURATION>  
<NBHOST>0</NBHOST>  
<SUBSCRIPTION_ID>237630</SUBSCRIPTION_ID>  
<Option_PROFILE_TITLE>Initial Options</Option_PROFILE_TITLE>  
<IS_PROCESSED>1</IS_PROCESSED>  
</SCAN_LAUNCH_INFO>  
</SCHEDULE_RUN>  
<SCAN_LAUNCH_INFO>  
<SCAN_ID>99553</SCAN_ID>  
<SCAN_REFERENCE>scan/1635431408.99553</SCAN_REFERENCE>  
<LAUNCH_DATETIME>2021-10-28 14:30:08</LAUNCH_DATETIME>  
<TITLE>test9090</TITLE>  
<TARGET>10.20.30.40</TARGET>  
<STATUS>FINISHED</STATUS>  
<DURATION>79 Seconds</DURATION>  
<NBHOST>0</NBHOST>  
<SUBSCRIPTION_ID>237630</SUBSCRIPTION_ID>  
<Option_PROFILE_TITLE>Initial Options</Option_PROFILE_TITLE>  
<IS_PROCESSED>1</IS_PROCESSED>  
</SCAN_LAUNCH_INFO>  
</SCHEDULE_RUN>  
<SCAN_LAUNCH_INFO>  
<SCAN_ID>99554</SCAN_ID>  
<SCAN_REFERENCE>scan/1635431408.99554</SCAN_REFERENCE>  
<LAUNCH_DATETIME>2021-10-28 14:30:08</LAUNCH_DATETIME>  
<TITLE>test898</TITLE>  
<TARGET>10.20.30.40</TARGET>  
<STATUS>FINISHED</STATUS>  
<DURATION>9 Seconds</DURATION>  
<NBHOST>0</NBHOST>  
<SUBSCRIPTION_ID>237630</SUBSCRIPTION_ID>  
<Option_PROFILE_TITLE>Initial Options</Option_PROFILE_TITLE>  
<IS_PROCESSED>1</IS_PROCESSED>  
</SCAN_LAUNCH_INFO>  
</SCHEDULE_RUN>  
</SCHEDULE_RUN_HISTORY>  
</SCHEDULE>
Compliance Scans

DTD
<platform_API_server>/api/2.0/fo/scan/schedules/runhistory/output.dtd

SCAP Scan List
/api/2.0/fo/scan/scap/ with action=list

[GET] [POST]

List SCAP scans in your account. By default the XML output lists scans launched in the past 30 days.

The input parameters for requesting a SCAP scan list are below. See Scan List Parameters for complete details.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=list (required), echo_request</td>
</tr>
<tr>
<td>Scan List Filters</td>
<td>scan_id (compliance scan ID), scan_ref, state, type, target, user_login, launched_afterDatetime, launched_before_datetime</td>
</tr>
<tr>
<td>Show Information</td>
<td>show_ags, show_op, show_status, show_last</td>
</tr>
</tbody>
</table>

API request 1: all SCAP scans

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list" "https://qualysapi.qualys.com/api/2.0/fo/scan/scap/"

API request 2: SCAP scan by reference number

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list&scan_ref=qscap/1402642816.80342" "https://qualysapi.qualys.com/api/2.0/fo/scan/scap/"

API request 3: On Demand SCAP scans only

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list&type=On-Demand" "https://qualysapi.qualys.com/api/2.0/fo/scan/scap/"

XML output:

<?xml version="1.0" encoding="UTF-8"?>
![DOCTYPE SCAN_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/scan/scap/qscap_scan_list_output.dtd">
<SCAN_LIST_OUTPUT>
  <RESPONSE>
    <SCAN_LIST>
      <SCAN>
        <ID>6980366</ID>
        <REF>qscap/1402694682.80366</REF>
        <TYPE>On-Demand</TYPE>
        <TITLE><![CDATA[<IMG SRC="http://www.google.com/images/logos/ps_logo2.png">]]></TITLE>
        <POLICY>
          <ID>39298</ID>
          <TITLE><![CDATA[Policy A]]></TITLE>
        </POLICY>
        <USER_LOGIN>acme_ab</USER_LOGIN>
        <STATUS>
          <STATE>Finished</STATE>
        </STATUS>
        <TARGET><![CDATA[10.10.30.244, 10.10.34.222]]></TARGET>
      </SCAN>
    </SCAN_LIST>
  </RESPONSE>
</SCAN_LIST_OUTPUT>

**DTD:**

`<platform API server>/api/2.0/fo/scan/qscap_scan_list_output.dtd`
Launch Compliance Scan

/api/2.0/fo/scan/compliance/?action=launch

[POST]

Launch compliance scan in the user’s account.

Using networks? Choose the Global Default Network to scan IPs on your network perimeter.

Input Parameters

The input parameters for launching a compliance scan are shown below. See Securing Amazon Web Services with Qualys

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=launch (required), echo_request, runtime_http_header</td>
</tr>
<tr>
<td>Scan Title</td>
<td>scan_title</td>
</tr>
<tr>
<td>Option Profile</td>
<td>option_id or option_title</td>
</tr>
<tr>
<td>Scanner Appliance</td>
<td>isscanner_id or isscanner_name</td>
</tr>
<tr>
<td>Asset IPs/Groups</td>
<td>ip, asset_group_ids, asset_groups, exclude_ip_per_scan, default_scanner, scanners_in_ag</td>
</tr>
<tr>
<td>Asset Tags</td>
<td>target_from=tags, use_ip_nt_range_tags_include, use_ip_nt_range_tags_exclude, use_ip_nt_range_tags, tag_include_selector, tag_exclude_selector, tag_set_by, tag_set_exclude, tag_set_include</td>
</tr>
<tr>
<td>Network</td>
<td>ip_network_id (when the Network Support feature is enabled)</td>
</tr>
<tr>
<td>Client</td>
<td>client_id and client_name (only for Consultant type subscriptions)</td>
</tr>
</tbody>
</table>

Sample - Launch a Compliance Scan

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=launch&ip=10.10.25.52&iscanner_name=iscan_er5&option_title=Initial+PC+Options&echo_request=1" "https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/" > apiOutputScan.txt
Sample - Launch a compliance scan using all scanners in network

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo 2" -d "action=launch&scan_title=pc+scan+API&option_id=3262&ip_network_id =12807913&scanners_in_network=1&ip=10.10.10.10,10.10.10.11" "https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <TEXT>New compliance scan launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>18198</VALUE>
      </ITEM>
      <ITEM>
        <KEY>REFERENCE</KEY>
        <VALUE>compliance/1473976536.18198</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Launch Compliance Scan on EC2 assets

/api/2.0/fo/scan/compliance/?action=launch

[POST]

Launch a compliance scan on your Amazon EC2 hosts (in your Amazon Web Services account).

A few things to consider...
- EC2 Scanning must be enabled for your Qualys account.
- Managers and Unit Managers can launch EC2 scans.
- Before scanning you’ll need to complete some set up steps. See Securing Amazon Web Services with Qualys
**Input Parameters**

The input parameters for launching an EC2 scan are shown below. Please see Scan Parameters for complete details.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=launch (required), echo_request</td>
</tr>
<tr>
<td>Scan Title</td>
<td>scan_title</td>
</tr>
<tr>
<td>EC2 environment</td>
<td>connector_name (required), ec2_endpoint (required)</td>
</tr>
<tr>
<td>Option Profile</td>
<td>option_id or option_title</td>
</tr>
<tr>
<td>Scanner Appliance</td>
<td>iscanner_id or iscanner_name</td>
</tr>
<tr>
<td>Target Hosts</td>
<td>target_from=tags (required)</td>
</tr>
</tbody>
</table>

Use tags to select the EC2 hosts you want to scan.

These parameters provide separate options for including and excluding tags for network IP ranges.

use_ip_nt_range_tags_include={0|1} (default in bold)

Important - This cannot be set to “1” for EC2 scanning.

use_ip_nt_range_tags_exclude={0|1} (default in bold)

Important - This cannot be set to “1” for EC2 scanning.

This parameter has been replaced with the include/exclude options above but it is still supported.

use_ip_nt_range_tags={0|1} (default in bold)

Important - This cannot be set to “1” for EC2 scanning.

These tag parameters are used to select tags:

- tag_set_include={tag1,tag2,...} (required)
- tag_set_exclude={tag1,tag2,...} (optional)
- tag_include_selector={any|all} (default in bold)
- tag_exclude_selector={any|all} (default in bold)
- tag_set_by={id|name} (default in bold)

**Sample - Launch EC2 compliance scan**

**API request:**

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
  -d
  "action=launch&scan_title=My+EC2+Scan+via+API&connector_name=EC2-Connector-Lab&ec2_endpoint=us-east-1&target_from=tags&tag_include_selector=any&tag_set_by=id&tag_set_include=270325&option_id=61769&iscanner_name=my-ec2-scanner"
  "https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
  "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
```
<REQUEST>
    <DATETIME>2018-06-24T10:10:51Z</DATETIME>
    <USER_LOGIN>USERNAME</USER_LOGIN>
    <RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/</RESOURCE>
</REQUEST>

<RESPONSE>
    <DATETIME>2018-06-24T10:10:57Z</DATETIME>
    <TEXT>New compliance scan launched</TEXT>
    <ITEM_LIST>
        <ITEM>
            <KEY>ID</KEY>
            <VALUE>2222345</VALUE>
        </ITEM>
        <ITEM>
            <KEY>REFERENCE</KEY>
            <VALUE>compliance/1347771234.36444</VALUE>
        </ITEM>
    </ITEM_LIST>
</RESPONSE>

Manage Compliance Scans

/api/2.0/fo/scan/compliance/?action={action}

Take actions on compliance scans in their account, like cancel, pause, resume, delete and fetch completed scan results.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) One action required for the request: cancel, pause, resume, delete, fetch</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_ref={value}</td>
<td>(Required) The scan reference for a compliance scan. This will have the format: compliance/nnnnnnnnnn.nnnnnn</td>
</tr>
</tbody>
</table>
Sample - Fetch PC Scan Results

API request:
```
curl -u USERNAME:PASSWORD -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/?
action=fetch&scan_ref=compliance/1347709693.37303" >
apiOutputScanFetch.txt
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE COMPLIANCE_SCAN_RESULT_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/compliance_scan_result_output.dtd">
<COMPLIANCE_SCAN_RESULT_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-06-17T10:23:53Z</DATETIME>
    <COMPLIANCE_SCAN>
      <HEADER>
        <NAME><![CDATA[Compliance Scan Results]]></NAME>
        <GENERATION_DATETIME>2012-09-17T10:23:53Z</GENERATION_DATETIME>
      </HEADER>
      <COMPANY_INFO>
        <NAME><![CDATA[Qualys]]></NAME>
        <ADDRESS><![CDATA[1600 Bridge Parkway]]></ADDRESS>
        <CITY><![CDATA[Redwood Shores]]></CITY>
        <STATE><![CDATA[California]]></STATE>
        <COUNTRY><![CDATA[United States]]></COUNTRY>
        <ZIP_CODE><![CDATA[94065]]></ZIP_CODE>
      </COMPANY_INFO>
      <USER_INFO>
        <NAME><![CDATA[NAME]]></NAME>
        <USERNAME>USERNAME</USERNAME>
        <ROLE>Manager</ROLE>
      </USER_INFO>
      <KEY value="USERNAME">USERNAME</KEY>
      <KEY value="COMPANY">Qualys</KEY>
      <KEY value="DATE">2018-06-15T11:49:08Z</KEY>
      <KEY value="TITLE">My PC Scan</KEY>
      <KEY value="TARGET">10.10.29</KEY>
      <KEY value="EXCLUDED_TARGET">N/A</KEY>
      <KEY value="DURATION">00:01:00</KEY>
      <KEY value="SCAN_HOST">10.10.21.122 (Scanner 6.6.28-1,
Vulnerability Signatures 2.2.215-2)</KEY>
      <KEY value="NBHOST_ALIVE">1</KEY>
      <KEY value="NBHOST_TOTAL">1</KEY>
    </COMPLIANCE_SCAN>
  </RESPONSE>
</COMPLIANCE_SCAN_RESULT_OUTPUT>
```
<KEY value="REPORT_TYPE">Scheduled</KEY>

<KEY value="OPTIONS">File Integrity Monitoring: Enabled, Scanned Ports: Standard Scan, Hosts to Scan in Parallel - External Scanners: 15, Hosts to Scan in Parallel - Scanner Appliances: 30, Total Processes to Run in Parallel: 10, HTTP Processes to Run in Parallel: 10,
Packet (Burst) Delay: Medium, Intensity: Normal, Overall Performance: Normal, ICMP Host Discovery, Ignore RST packets: Off, Ignore firewall-generated SYN-ACK packets: Off, Do not send ACK or SYN-ACK packets during host discovery: Off</KEY>

<KEY value="STATUS">FINISHED</KEY>

<OPTION_PROFILE>
  <OPTION_PROFILE_TITLE option_profile_default="0">![CDATA[11412]]</OPTION_PROFILE_TITLE>
</OPTION_PROFILE>

</HEADER>

<APPENDIX>
  <TARGET_HOSTS>
    <HOSTS_SCANNED>10.10.10.29</HOSTS_SCANNED>
  </TARGET_HOSTS>
  <TARGET_DISTRIBUTION>
    <SCANNER>
      <NAME>![CDATA[iscan_sx]]></NAME>
      <HOSTS>10.10.10.29</HOSTS>
    </SCANNER>
  </TARGET_DISTRIBUTION>
  <AUTHENTICATION>
    <AUTH>
      <TYPE>Windows</TYPE>
      <SUCCESS>
        <IP>10.10.10.29</IP>
      </SUCCESS>
    </AUTH>
  </AUTHENTICATION>
</APPENDIX>
</COMPLIANCE_SCAN>
</RESPONSE>
</COMPLIANCE_SCAN_RESULT_OUTPUT>
Cloud Perimeter Scans

/api/2.0/fo/scan/cloud/perimeter/job/

[POST]

Cloud perimeter scans are available for VM and PC modules. Only Managers and Unit Managers have permission to configure cloud perimeter scans.

The input parameters for requesting a Cloud Perimeter scan are below. See Cloud Perimeter Scan Parameters for complete details.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action={create</td>
</tr>
<tr>
<td>Scan List Filters</td>
<td>id, module, cloud_provider, cloud_service,</td>
</tr>
<tr>
<td></td>
<td>connector_name, connector_uuid, scan_title, active,</td>
</tr>
<tr>
<td></td>
<td>option_title, option_id, priority, scanner_id,</td>
</tr>
<tr>
<td></td>
<td>isscanner_name, platform_type, region_code, vpc_id,</td>
</tr>
<tr>
<td></td>
<td>tag_include_selector, tag_exclude_selector, tag_set_by,</td>
</tr>
<tr>
<td></td>
<td>tag_set_include, tag_set_exclude, elb_dns, schedule</td>
</tr>
</tbody>
</table>

Create/Update Cloud Perimeter Scan

We allow you to create/update a cloud perimeter scan job through Cloud Perimeter Scan API even if no scan targets are resolved from the provided details. At the time of scan, if no scan targets are resolved from the provided details, the scan will not be launched, and we add the error in the Activity log and Run history of the schedule scan job.

API Request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"action=create&tag_set_by=name&tag_include_selector=any&tag_set_include=ec2-Virginia,Unassigned Business Unit&connector_name=conn1&region_code=us-east-1&active=1&option_title=Initial Options&module=vm&schedule=now&cloud_provider=aws&platform_type=classic&after_notify=1&after_notify_message=Scan Finished"
"https://qualysapi.qualys.com/api/2.0/fo/scan/cloud/perimeter/job/"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-11T04:06:01Z</DATETIME>
    <TEXT>Scan has been created successfully</TEXT>
    <ITEM_LIST>
Example - Create Cloud Perimeter Scan Job (Recurring Schedule)

**API Request:**

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"action=create&tag_set_by=name&tag_include_selector=any&tag_set_in
clude=EC2_Targets&tag_exclude_selector=any&tag_set_exclude=EC2_Tes
t&connector_name=EC2 Connector&region_code=us-east-1&active=0&occurrence=daily&start_date=04/02/2018&start_hour=10&st
art_minute=30&time_zone_code=IN&option_title=Initial
Options&frequency_days=364&observe_dst=no&module=vm&schedule=recur
ring&cloud_provider=aws&platform_type=classic&after_notify=1&recip
ient_group_ids=4229"
"https://qualysapi.qualys.com/api/2.0/fo/scan/cloud/perimeter/job/"

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
 <RESPONSE>
  <DATETIME>2018-04-11T05:01:42Z</DATETIME>
  <TEXT>Scan has been created successfully</TEXT>
  <ITEM_LIST>
   <ITEM>
    <KEY>ID</KEY>
    <VALUE>1352071</VALUE>
   </ITEM>
  </ITEM_LIST>
 </RESPONSE>
</SIMPLE_RETURN>
```

Example - Update Cloud Perimeter Scan Job

**API Request:**

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"action=update&id=1352071&connector_name=EC2Connector-2&platform_type=vpc_peered&region_code=us-west-1"
Scans
Cloud Perimeter Scans

"https://qualysapi.qualys.com/api/2.0/fo/scan/cloud/perimeter/job/"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-11T05:05:35Z</DATETIME>
    <TEXT>Scan has been updated successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>1352071</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

DTD

```xml
<platform API server>/api/2.0/fo/scan/simple_return.dtd
```
VM Scan Schedules

The Schedule Scan API (/api/2.0/fo/schedule/scan/) is used to define schedules for vulnerability scans in the user’s account.

Permissions

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>Create scan schedules for all assets in the subscription</td>
</tr>
<tr>
<td></td>
<td>Remove all scan schedules</td>
</tr>
<tr>
<td></td>
<td>View all scan schedules in the subscription</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>Create scan schedules for assets in user’s business unit</td>
</tr>
<tr>
<td></td>
<td>Remove scan schedules in user’s business unit.</td>
</tr>
<tr>
<td></td>
<td>View scan schedules in the subscription*</td>
</tr>
<tr>
<td>Scanner</td>
<td>Create scan schedules for assets in user’s account.</td>
</tr>
<tr>
<td></td>
<td>Remove user’s scan schedules</td>
</tr>
<tr>
<td></td>
<td>View scan schedules in the subscription*</td>
</tr>
<tr>
<td>Readers</td>
<td>No permission to create or remove scan schedules</td>
</tr>
<tr>
<td></td>
<td>View scan schedules in the subscription*</td>
</tr>
</tbody>
</table>

* Qualys includes an account permission setting that restricts Unit Managers, Scanners, and Readers from viewing scheduled tasks on unassigned assets.

List scan schedules

/api/2.0/fo/schedule/scan/?action=list

[GET] [POST]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Optional) The ID of the scan schedule you want to display.</td>
</tr>
<tr>
<td>active={0</td>
<td>1}</td>
</tr>
<tr>
<td>show_notifications={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_type=certview</td>
<td>(Optional) Launch a CertView type VM scan. This option will be supported when CertView GA is released and enabled for your account.</td>
</tr>
<tr>
<td>scan_type=ec2certview</td>
<td>(Optional) Launch a CertView type VM scan for EC2 assets.</td>
</tr>
</tbody>
</table>
Scans
VM Scan Schedules

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule(scan/?action=list&id=160642&show_notifications=1"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCHEDULE_SCAN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/schedule(scan/schedule_scan_list_output.dtd">
<SCHEDULE_SCAN_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-12-01T19:26:50Z</DATETIME>
    <SCHEDULE_SCAN_LIST>
      <SCAN>
        <ID>160642</ID>
        <ACTIVE>1</ACTIVE>
        <TITLE><![CDATA[My Daily Scan]]></TITLE>
        <USER_LOGIN>qualys_ps</USER_LOGIN>
        <TARGET><![CDATA[10.10.10.10-10.10.10.20]]></TARGET>
        <NETWORK_ID><![CDATA[0]]></NETWORK_ID>
        <ISCANNER_NAME><![CDATA[External Scanner]]></ISCANNER_NAME>
        <USER_ENTERED_IPS>
          <RANGE>
            <START>10.10.10.10</START>
          </RANGE>
        </USER_ENTERED_IPS>
      </SCAN>
    </SCHEDULE_SCAN_LIST>
  </RESPONSE>
</SCHEDULE_SCAN_LIST_OUTPUT>

Parameter | Description
--- | ---
fqdn={value} | (Optional) The target FQDN for a vulnerability scan. You must specify at least one target i.e. IPs, asset groups or FQDNs. Multiple values are comma separated.
show_cloud_details={0|1} | (Optional) Set to 1 to display the cloud details (Provider, Connector, Scan Type and Cloud Target) in the XML output. Otherwise the details are not displayed in the output.
client_id={value} | (Optional) Id assigned to the client (Consultant type subscription only). Parameter client_id or client_name may be specified for the same request.
client_name={value} | (Optional) Name of the client (Consultant type subscription only). Parameter client_id or client_name may be specified for the same request.
scan_type=perimeter | (Optional) List cloud perimeter scans only. This option will be supported for Cloud Perimeter Scans in future release.
show_cloud_details={0|1} | (Optional) Set to 1 to display cloud details in the XML output. The cloud details will show scan type "Cloud Perimeter" for cloud perimeter scans.

Parameter

description

fqdn={value} (Optional) The target FQDN for a vulnerability scan. You must specify at least one target i.e. IPs, asset groups or FQDNs. Multiple values are comma separated.
show_cloud_details={0|1} (Optional) Set to 1 to display the cloud details (Provider, Connector, Scan Type and Cloud Target) in the XML output. Otherwise the details are not displayed in the output.
client_id={value} (Optional) Id assigned to the client (Consultant type subscription only). Parameter client_id or client_name may be specified for the same request.
client_name={value} (Optional) Name of the client (Consultant type subscription only). Parameter client_id or client_name may be specified for the same request.
scan_type=perimeter (Optional) List cloud perimeter scans only. This option will be supported for Cloud Perimeter Scans in future release.
show_cloud_details={0|1} (Optional) Set to 1 to display cloud details in the XML output. The cloud details will show scan type "Cloud Perimeter" for cloud perimeter scans.

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule(scan/?action=list&id=160642&show_notifications=1"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCHEDULE_SCAN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/schedule(scan/schedule_scan_list_output.dtd">
<SCHEDULE_SCAN_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-12-01T19:26:50Z</DATETIME>
    <SCHEDULE_SCAN_LIST>
      <SCAN>
        <ID>160642</ID>
        <ACTIVE>1</ACTIVE>
        <TITLE><![CDATA[My Daily Scan]]></TITLE>
        <USER_LOGIN>qualys_ps</USER_LOGIN>
        <TARGET><![CDATA[10.10.10.10-10.10.10.20]]></TARGET>
        <NETWORK_ID><![CDATA[0]]></NETWORK_ID>
        <ISCANNER_NAME><![CDATA[External Scanner]]></ISCANNER_NAME>
        <USER_ENTERED_IPS>
          <RANGE>
            <START>10.10.10.10</START>
          </RANGE>
        </USER_ENTERED_IPS>
      </SCAN>
    </SCHEDULE_SCAN_LIST>
  </RESPONSE>
</SCHEDULE_SCAN_LIST_OUTPUT>
<END>10.10.10.20</END>
</RANGE>
</USER_ENTERED_IPS>
</OPTION_PROFILE>
</PROCESSING_PRIORITY>
</SCHEDULE>
</SCHEDULE_SCAN_LIST>
</RESPONSE>
</SCHEDULE_SCAN_LIST_OUTPUT>
Example: Users can filter the schedule scan list to only show cloud perimeter scan jobs. Also, when you include cloud details in the output, we’ll show scan type “Cloud Perimeter”.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule(scan/?action=list&id=1340788&scan_type=perimeter&show_cloud_details=1"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCHEDULE_SCAN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/schedule(scan scheduler_scan_list_output.dtd">
<SCHEDULE_SCAN_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2018-04-12T12:57:03Z</DATETIME>
        <SCHEDULE_SCAN_LIST>
            <SCAN>
                <ID>1340788</ID>
                <ACTIVE></ACTIVE>
                <TITLE><![CDATA[My_External_Scan]]></TITLE>
                <USER_LOGIN>utwrx_mp</USER_LOGIN>
                <TARGET><![CDATA[Asset Tags Included]]></TARGET>
                <ISCANNER_NAME><![CDATA[External Scanner]]></ISCANNER_NAME>
                <EC2_INSTANCE>
                    <CONNECTOR_UUID><![CDATA[8047abce-c3ac-42e0-ad49-be4181d22c84]]></CONNECTOR_UUID>
                    <EC2_ENDPOINT><![CDATA[1507b6c1-07a7-4d88-acf2-8c6b63e749c4]]></EC2_ENDPOINT>
                    <EC2_ONLY_CLASSIC><![CDATA[1]]></EC2_ONLY_CLASSIC>
                </EC2_INSTANCE>
                <CLOUD_DETAILS>
                    <PROVIDER>AWS</PROVIDER>
                    <CONNECTOR>
                        <ID>37361</ID>
                        <UUID>8047abce-c3ac-42e0-ad49-be4181d22c84</UUID>
                        <NAME><![CDATA[EC2 Connector]]></NAME>
                    </CONNECTOR>
                    <SCAN_TYPE>Cloud Perimeter</SCAN_TYPE>
                    <CLOUD_TARGET>
                        <PLATFORM>Classic</PLATFORM>
                        <REGION>
                            <UUID>1507b6c1-07a7-4d88-acf2-8c6b63e749c4</UUID>
                            <CODE>us-east-1</CODE>
                            <NAME><![CDATA[US East (N. Virginia)]]></NAME>
                        </REGION>
                    </CLOUD_TARGET>
                </CLOUD_DETAILS>
            </SCAN>
        </SCHEDULE_SCAN_LIST>
    </RESPONSE>
</SCHEDULE_SCAN_LIST_OUTPUT>
```
<REGION>
  <VPC_SCOPE>None</VPC_SCOPE>
</CLOUD_TARGET>
</CLOUD_DETAILS>
<ASSET_TAGS>
  <TAG_INCLUDE_SELECTOR>any</TAG_INCLUDE_SELECTOR>
  <TAG_SET_INCLUDE><![CDATA[EC2_Targets]]></TAG_SET_INCLUDE>
  <TAG_EXCLUDE_SELECTOR>any</TAG_EXCLUDE_SELECTOR>
  <TAG_SET_EXCLUDE><![CDATA[EC2_Test]]></TAG_SET_EXCLUDE>
  <USE_IP_NT_RANGE_TAGS>0</USE_IP_NT_RANGE_TAGS>
</ASSET_TAGS>
<ELB_DNS>
  <DNS><![CDATA[abc.com]]></DNS>
  <DNS><![CDATA[abc123.com]]></DNS>
</ELB_DNS>
<OPTION_PROFILE>
  <TITLE><![CDATA[Initial Options]]></TITLE>
  <DEFAULT_FLAG>1</DEFAULT_FLAG>
</OPTION_PROFILE>
<PROCESSING_PRIORITY>0 - No Priority</PROCESSING_PRIORITY>
<SCHEDULE>
  <DAILY frequency_days="364" />
  <START_DATE_UTC>2018-04-02T05:00:00Z</START_DATE_UTC>
  <START_HOUR>10</START_HOUR>
  <START_MINUTE>30</START_MINUTE>
  <TIME_ZONE>
    <TIME_ZONE_CODE>IN</TIME_ZONE_CODE>
    <TIME_ZONE_DETAILS>(GMT+0530) India: Asia/Calcutta</TIME_ZONE_DETAILS>
  </TIME_ZONE>
  <DST_SELECTED>0</DST_SELECTED>
</SCHEDULE>
</SCAN>
</SCHEDULE_SCAN_LIST>
</RESPONSE>
</SCHEDULE_SCAN_LIST_OUTPUT>

**DTD**

<platform API server>/api/2.0/fo/schedule(scan/schedule_scan_list_output.dtd
Create scan schedule

/api/2.0/fo/schedule/scan/?action=create

[POST]

Create a scan schedule in the user’s account.

**Input Parameters**

The input parameters for creating a scan schedule are below. For complete details see Scan Parameters and Scan Schedule Parameters.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=create (required), echo_request</td>
</tr>
<tr>
<td>Scan</td>
<td>scan_title (required), active=0</td>
</tr>
<tr>
<td>Option Profile</td>
<td>option_id or option_title (one is required)</td>
</tr>
<tr>
<td>Scanner Appliance</td>
<td>isscanner_id or isscanner_name</td>
</tr>
<tr>
<td>Processing Priority</td>
<td>priority</td>
</tr>
<tr>
<td>Asset IPs/Groups</td>
<td>ip, asset_group_ids, asset_groups, exclude_ip_per_scan, default_scanner, scanners_in_ag</td>
</tr>
<tr>
<td>Asset Tags</td>
<td>target_from=tags, tag_include_selector, tag_exclude_selector, tag_set_by, tag_set_exclude, tag_set_include, use_ip_nt_range_tags_include, use_ip_nt_range_tags_exclude, use_ip_nt_range_tags</td>
</tr>
<tr>
<td>Network</td>
<td>ip_network_id to filter IPs/ranges in “ip” parameter (valid when the networks feature is enabled)</td>
</tr>
<tr>
<td>EC2 Hosts</td>
<td>target_from=tags (required)</td>
</tr>
<tr>
<td></td>
<td>use_ip_nt_range_tags_include=0 (optional)</td>
</tr>
<tr>
<td></td>
<td>use_ip_nt_range_tags_exclude=0 (optional)</td>
</tr>
<tr>
<td></td>
<td>use_ip_nt_range_tags=0 (optional)</td>
</tr>
<tr>
<td></td>
<td>tag_set_include (required)</td>
</tr>
<tr>
<td></td>
<td>More Asset Tags parameters (optional)</td>
</tr>
<tr>
<td>EC2 Environment</td>
<td>connector_name or connector_uuid (one is required)</td>
</tr>
<tr>
<td></td>
<td>ec2_endpoint (required)</td>
</tr>
<tr>
<td>Scheduling</td>
<td>start_date (current date by default)</td>
</tr>
<tr>
<td></td>
<td>start_hour, start_minute, time_zone_code, occurrence (required)</td>
</tr>
<tr>
<td></td>
<td>observe_dst, recurrence, end_after, pause_after_hours, resume_in_days</td>
</tr>
<tr>
<td>Daily Scan</td>
<td>occurrence=daily, frequency_days (required)</td>
</tr>
<tr>
<td>Weekly Scan</td>
<td>occurrence=weekly, frequency_weeks, weeks (required)</td>
</tr>
</tbody>
</table>
**Type** | **Parameter List**
---|---
Monthly Scan | occurrence=monthly, frequency_months (required)
 | Nth day of month: day_of_month (required)
 | Day in Nth week: day_of_week, week_of_month (required)
Notifications | before_notify, before_notify_unit, before_notify_time,
 | before_notify_message, after_notify, after_notify_message,
 | recipient_group_ids, delay_notify, delay_notify_message,
 | skipped_notify, skipped_notify_message, deactivate_notify,
 | deactivate_notify_message

**Sample - Create scan schedule**

**API request:**
```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -X "POST" -d "scan_title=My+Scan+Schedule&active=1&option_id=3456&target_from=tags&tag_set_include=tag1,tag2,tag3&iscanner_name=scanner1&occurrence=daily&frequency_days=5&time_zone_code=US-CA&observe_dst=yes&start_hour=14&start_minute=0"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/?action=create"
```

**Sample - Create Scan Schedule, Cancel after 45 minutes**

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=create&scan_title=My_Weekly_Scan&option_title=InitialOptions&ip=10.20.31.73,10.20.31.106&active=1&occurrence=weekly&start_hour=13&start_minute=30&time_zone_code=IN&frequency_weeks=1&weekdays=Sunday&end_after=0&end_after_mins=45&iscanner_name=scanner1,scanner2&before_notify=1&before_notify_unit=hours&before_notify_time=20&recipient_group_ids=4228,5628"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/"
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
 <RESPONSE>
  <DATETIME>2019-01-02T21:32:40Z</DATETIME>
  <TEXT>New scan scheduled successfully</TEXT>
  <ITEM_LIST>
   <ITEM>
    <KEY>ID</KEY>
    <VALUE>136992</VALUE>
   </ITEM>
  </ITEM_LIST>
 </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Create scan schedule using all scanners in network

**API request:**

curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl demo 2" -d "action=create&scan_title=API+Schedule+scan&option_title=Initial+Options&ip_network_id=12807913&scanners_in_network=1&ip=10.10.10.10,10.10.10.11&occurrence=monthly&frequency_months=12&day_of_month=20&start_minute=00&start_hour=22&time_zone_code=IN&observe_dst=no&pause_after_hours=3&resume_in_days=4&recurrence=5&start_date=08/20/2016&active=1" "https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/"

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
<RESPONSE>
<DATETIME>2018-04-20T21:32:40Z</DATETIME>
<TEXT>New scan scheduled successfully</TEXT>
<ITEM_LIST>
<ITEM>
  <KEY>ID</KEY>
  <VALUE>136992</VALUE>
</ITEM>
</ITEM_LIST>
</RESPONSE>
</SIMPLE_RETURN>
```

Sample - Create scan schedule (with FQDN and asset group)

**API request:**

curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=create&scan_title=My+Schedule&active=1&time_zone_code=US-OR&start_hour=18&start_minute=50&occurrence=daily&option_title=Initial+Options&frequency_days=1&asset_groups=My+AG&fqdn=domain.qualys.com" "https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/"
Update a scan schedule

/api/2.0/fo/schedule(scan/?action=update

[POST]

Update a scan schedule in the user’s account. During an update request you must specify target_from=assets when fqdn is specified in the same request. This is true for vulnerability scans and CertView type vulnerability scans.

When fqdn is not specified during an update request for a scheduled scan that already has fqdn defined, we will keep the existing value.

Input Parameters

The input parameters for updating a scan schedule are below. For complete details see Scan Parameters and Scan Schedule Parameters.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=update (required), id (required), echo_request</td>
</tr>
<tr>
<td>Scan Title</td>
<td>scan_title</td>
</tr>
<tr>
<td>Status</td>
<td>active=0</td>
</tr>
<tr>
<td>Option Profile</td>
<td>option_id or option_title</td>
</tr>
<tr>
<td>Scanner Appliance</td>
<td>iscanner_id, iscanner_name, default_scanner, scanners_in_ag, scanners_in_network, scanners_in_tagset</td>
</tr>
<tr>
<td>Processing Priority</td>
<td>priority</td>
</tr>
<tr>
<td>Asset IPs/Groups</td>
<td>ip, asset_group_ids or asset_groups, exclude_ip_per_scan</td>
</tr>
<tr>
<td>Asset Tags</td>
<td>target_from=tags, use_ip_nt_range_tags_include, use_ip_nt_range_tags_exclude, use_ip_nt_range_tags, tag_include_selector, tag_exclude_selector, tag_set_by, tag_set_exclude, tag_set_include</td>
</tr>
<tr>
<td>EC2 Environment</td>
<td>connector_name or connector_uuid, ec2_endpoint, ec2_only_classic</td>
</tr>
<tr>
<td>Network</td>
<td>ip_network_id (when the Network Support feature is enabled)</td>
</tr>
<tr>
<td>Start Time</td>
<td>Must be specified together: set_start_time=1, start_date, start_hour, start_minute, time_zone_code, observe_dst</td>
</tr>
<tr>
<td>Recurrence</td>
<td>recurrence</td>
</tr>
<tr>
<td>Daily Scan</td>
<td>Must be specified together: occurrence=daily, frequency_days</td>
</tr>
<tr>
<td>Weekly Scan</td>
<td>Must be specified together: occurrence=weekly, frequency_weeks, weekdays</td>
</tr>
</tbody>
</table>
Scans
VM Scan Schedules

Sample - Update scan schedule, Pause after 15 minutes

API request:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=update&id=146754&pause_after_hours=0&pause_after_mins=15&resume_in_days=2&resume_in_hours=5"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/"
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" >
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
< SIMPLE_RETURN >
< RESPONSE >
  < DATETIME >2019-01-14T11:57:42Z</ DATETIME >
  < TEXT >Edit scheduled Scan Completed successfully</ TEXT >
  < ITEM_LIST >
    < ITEM >
      < KEY >ID</ KEY >
      < VALUE >146754</ VALUE >
    </ ITEM >
  </ ITEM_LIST >
</ RESPONSE >
</ SIMPLE_RETURN >
```
Delete scan schedule

/api/2.0/fo/schedule/scan/?action=update

[POST]

Delete a scan schedule in the user’s account.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0/1</td>
<td>(Optional) Specify 1 to echo the request’s input parameters (names and values) in the XML output. Otherwise parameters are not displayed in the output.</td>
</tr>
<tr>
<td>id=value</td>
<td>(Optional) The ID of the scan schedule you want to delete.</td>
</tr>
</tbody>
</table>

**Sample - Delete scan schedule**

**API request:**

```bash
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -X "POST" -d "id=123456"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/?action=delete"
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-05-30T21:32:40Z</DATETIME>
    <TEXT>Schedule scan deleted successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>123456</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
PC Scan Schedules

The PC Schedule Scan API (/api/2.0/fo/schedule.scan/compliance) allows you to create, update, list, and delete schedule scans for Policy Compliance.

Permissions

Note: The PC Scan schedule APIs are available as part of one of the following subscription combinations only:

- PC and API add-on
- PC, SCA, and API add-on
- VMDR, SCA, and API add-on

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>Create scan schedules for all assets in the subscription</td>
</tr>
<tr>
<td></td>
<td>Remove all scan schedules</td>
</tr>
<tr>
<td></td>
<td>View all scan schedules in the subscription</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>Create scan schedules for assets in user’s business unit</td>
</tr>
<tr>
<td></td>
<td>Remove scan schedules in user’s business unit.</td>
</tr>
<tr>
<td></td>
<td>View scan schedules in the subscription*</td>
</tr>
<tr>
<td>Scanner</td>
<td>Create scan schedules for assets in user’s account.</td>
</tr>
<tr>
<td></td>
<td>Remove user’s scan schedules</td>
</tr>
<tr>
<td></td>
<td>View scan schedules in the subscription*</td>
</tr>
<tr>
<td>Readers</td>
<td>No permission to create or remove scan schedules</td>
</tr>
<tr>
<td></td>
<td>View scan schedules in the subscription*</td>
</tr>
</tbody>
</table>

*Qualys includes an account permission setting that restricts Unit Managers, Scanners, and Readers from viewing scheduled tasks on unassigned assets.
List compliance scan schedules

/api/2.0/fo/schedule/scan/compliance/?action=list

[GET]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0</td>
<td>1 (Optional) Specify 1 to echo the request's input parameters (names and values) in the XML output. Otherwise parameters are not displayed in the output.</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Optional) The ID of the scan schedule you want to display.</td>
</tr>
<tr>
<td>active=0</td>
<td>1 (Optional) Specify 1 for active schedules only, or 0 for deactivated schedules only.</td>
</tr>
<tr>
<td>show_notifications=0</td>
<td>1 (Optional) Specify 1 to include the notification settings for each schedule in the XML output.</td>
</tr>
<tr>
<td>show_cloud_details=0</td>
<td>1 (Optional) Set to 1 to display the cloud details (Provider, Connector, Scan Type and Cloud Target) in the XML output. Otherwise the details are not displayed in the output.</td>
</tr>
<tr>
<td>client_id={value}</td>
<td>(Optional) Id assigned to the client (Consultant type subscription only). Parameter client_id or client_name may be specified for the same request.</td>
</tr>
<tr>
<td>client_name={value}</td>
<td>(Optional) Name of the client (Consultant type subscription only). Parameter client_id or client_name may be specified for the same request.</td>
</tr>
</tbody>
</table>

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/compliance/?action=list"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE COMPLIANCE_SCHEDULE_SCAN_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/compliance/compliance_schedule_scan_list_output.dtd">
<COMPLIANCE_SCHEDULE_SCAN_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2019-11-19T10:10:58Z</DATETIME>
        <COMPLIANCE_SCHEDULE_SCAN_LIST>
            <SCAN>
                <ID>57363</ID>
                <ACTIVE>1</ACTIVE>
                <TITLE>
```
<![CDATA[My Scan Schedule api6]]>
</TITLE>
<User_LOGIN>quays_sp1</User_LOGIN>
<TARGET>
  <![CDATA[10.10.10.185]]>
</TARGET>
<NEXT_LAUNCH_UTC>2019-11-19T22:00:00Z</NEXT_LAUNCH_UTC>
<TIME_ZONE>US-CA</TIME_ZONE>
</SCANNER_NAME>
</SCHEDULE>
</NOTIFICATIONS />
</SCAN>
</COMPLIANCE_SCHEDULE_SCAN_LIST>
</RESPONSE>
Create a Compliance Scan Schedule

/api/2.0/fo/schedule/scan/compliance/?action=create

[POST]

Create a scan schedule in the user’s account.

Input Parameters
The input parameters for creating a scan schedule are below. For complete details see Scan Parameters and Scan Schedule Parameters.

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=create (required),</td>
</tr>
<tr>
<td>echo_request</td>
<td>(Optional) Specify 1 to echo the request’s input parameters (names and values) in the XML output. Otherwise parameters are not displayed in the output.</td>
</tr>
<tr>
<td>Scan</td>
<td>scan_title (required), active=0/1 (required)</td>
</tr>
<tr>
<td>Compliance Profile</td>
<td>option_id or option_profile (one is required)</td>
</tr>
<tr>
<td>Scanner Appliance</td>
<td>iscanner_id or iscanner_name</td>
</tr>
<tr>
<td>Asset IPs/Groups</td>
<td>ip, asset_group_ids, asset_groups, exclude_ip_per_scan, default_scanner, scanners_in_ag</td>
</tr>
<tr>
<td>Asset Tags</td>
<td>target_from=tags, tag_include_selector,</td>
</tr>
<tr>
<td></td>
<td>tag_exclude_selector, tag_set_by, tag_set_exclude,</td>
</tr>
<tr>
<td></td>
<td>tag_set_include, use_ip_nt_range_tags_include, use_ip_nt_range_tags_exclude, use_ip_nt_range_tags</td>
</tr>
<tr>
<td>Network</td>
<td>ip_network_id to filter IPs/ranges in “ip” parameter (valid when the networks feature is enabled)</td>
</tr>
<tr>
<td>Scheduling</td>
<td>start_date (current date by default)</td>
</tr>
<tr>
<td></td>
<td>start_hour, start_minute, time_zone_code, occurrence (required)</td>
</tr>
<tr>
<td></td>
<td>observe_dst, recurrence, end_after, pause_after_hours, resume_in_days</td>
</tr>
<tr>
<td>Daily Scan</td>
<td>occurrence=daily, frequency_days (required)</td>
</tr>
<tr>
<td>Weekly Scan</td>
<td>occurrence=weekly, frequency_weeks, weeks (required)</td>
</tr>
</tbody>
</table>
Samples - Create compliance scan schedule

**API request:**
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/compliance/
?action=create&scan_title=My+Scan+Schedule+api6&active=1&option_id
=76960&asset_groups=policyred7&iscanner_name=pyscandsp&occurrence=
daily&frequency_days=5&time_zone_code=US-CA&observe_dst=yes&start_hour=14&start_minute=0"
```

**XML output:**
```xml
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
 <RESPONSE>
  <DATETIME>2019-11-19T11:14:19Z</DATETIME>
  <TEXT>New compliance scan scheduled successfully</TEXT>
  <ITEM_LIST>
   <ITEM>
    <KEY>ID</KEY>
    <VALUE>57368</VALUE>
   </ITEM>
  </ITEM_LIST>
 </RESPONSE>
</SIMPLE_RETURN>
```

Samples - Create compliance scan schedule and cancel after 45 minutes

**API request:**
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/compliance/
?action=create&scan_title=My_Weekly_Scan&option_title=nordea
windows&ip=10.10.10.10&active=1&occurrence=weekly&start_hour=13&start_minute=30&time_zone_code=IN&frequency_weeks=1&weekdays=Sunday&end_after=0&end_after_mins=45&iscanner_name=pyscandsp&before_notify=1&before_notify_unit=hours&before_notify_time=20"
```

**Monthly Scan occurrence=monthly, frequency_months (required) **
Nth day of month: day_of_month (required)
Day in Nth week: day_of_week, week_of_month (required)

**Notifications**
before_notify, before_notify_unit, before_notify_time,
before_notify_message, after_notify, after_notify_message,
recipient_group_ids, delay_notify, delay_notify_message,
 skipped_notify, skipped_notify_message, deactivate_notify,
deactivate_notify_message
XML output:

```xml
<DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2019-11-21T08:06:49Z</DATETIME>
    <TEXT>New compliance scan scheduled successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>57369</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Create compliance scan schedule using all scanners in network

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/compliance/
?action=create&scan_title=API+Schedule+scan&option_title=nordea
windows&ip_network_id=52010&scanners_in_network=1&ip=10.10.10.10
,10.10.10.11&occurrence=monthly&frequency_months=12&day_of_month=20
&start_minute=00&start_hour=22&time_zone_code=IN&observe_dst=no&pause_after_hours=3&resume_in_days=4&recurrence=5&start_date=08/20/2020&active=1"
```

XML output:

```xml
<DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2019-11-21T08:26:00Z</DATETIME>
    <TEXT>New compliance scan scheduled successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>57370</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Create EC2 compliance scan schedule

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" 
"https://qualysapi.qualys.com/api/2.0/fo/schedule(scan/compliance/?
action=create&scan_title=API_Schedule_EC2_PC&target_from=tags&tag_set_by=name&tag_include_selector=any&tag_set_include=Auth&connector_name=AWS+Connector&ec2_endpoint=us-east-1&active=0&occurrence=daily&start_date=05/21/2020&start_hour=20&start_minute=30&time_zone_code=IN&option_title=Initial+PC+Options&frequency_days=364&end_after=1&observe_dst=no&iscanner_name=EC2_Scan ner"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2020-06-07T22:09:26Z</DATETIME>
    <TEXT>New compliance scan scheduled successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>279256</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
Update a Compliance Scan Schedule

/api/2.0/fo/schedule/scan/compliance/?action=update&id=<id>

[POST]

Update a scan schedule in the user’s account.

**Input Parameters**

The input parameters for updating a scan schedule are below. For complete details see [Scan Parameters](#) and [Scan Schedule Parameters](#).

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>action=update (required)</td>
</tr>
<tr>
<td>echo_request=0</td>
<td>1) (Optional) Specify 1 to echo the request’s input parameters (names and values) in the XML output. Otherwise parameters are not displayed in the output.</td>
</tr>
<tr>
<td>Scan Title</td>
<td>scan_title</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) The ID of the scan schedule you want to update.</td>
</tr>
<tr>
<td>Status</td>
<td>active=0</td>
</tr>
<tr>
<td>Compliance Profile</td>
<td>option_id or option_title</td>
</tr>
<tr>
<td>Scanner Appliance</td>
<td>iscanner_id, iscanner_name, default_scanner,</td>
</tr>
<tr>
<td></td>
<td>scanners_in_ag, scanners_in_network, scanners_in_tagset</td>
</tr>
<tr>
<td>Asset IPs/Groups</td>
<td>ip, asset_group_ids or asset_groups, exclude_ip_per_scan</td>
</tr>
<tr>
<td>Asset Tags</td>
<td>target_from=tags, use_ip_nt_range_tags_include,</td>
</tr>
<tr>
<td></td>
<td>use_ip_nt_range_tags_exclude, use_ip_nt_range_tags, tag_include_selector,</td>
</tr>
<tr>
<td></td>
<td>tag_exlude_selector, tag_set_by, tag_set_exclude, tag_set_include</td>
</tr>
<tr>
<td>Network</td>
<td>ip_network_id (when the Network Support feature is enabled)</td>
</tr>
<tr>
<td>Start Time</td>
<td>Must be specified together:</td>
</tr>
<tr>
<td></td>
<td>set_start_time=1, start_date, start_hour, start_minute, time_zone_code,</td>
</tr>
<tr>
<td></td>
<td>observe_dst</td>
</tr>
<tr>
<td>recurrence={value}</td>
<td>(Optional) The number of times the scan will be run before it is deactivated. For example, if you set recurrence=2, the scan schedule will be deactivated after it runs 2 times. By default no value is set. A valid value is an integer from 1 to 99.</td>
</tr>
<tr>
<td>Daily Scan</td>
<td>Must be specified together:</td>
</tr>
<tr>
<td></td>
<td>occurrence=daily, frequency_days</td>
</tr>
<tr>
<td>Weekly Scan</td>
<td>Must be specified together:</td>
</tr>
<tr>
<td></td>
<td>occurrence=weekly, frequency_weeks, weekdays</td>
</tr>
</tbody>
</table>
Sample - Update compliance scan schedule.

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"http://qualysapi.qualys.com/api/2.0/fo/schedule/scan/compliance/?
action=update&id=57360&option_id=39594"

XML output:

```xml
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2019-11-19T12:04:44Z</DATETIME>
    <TEXT>Edit scheduled Scan Completed successfully</TEXT>
    <ITEM_LIST>
    <ITEM>
      <KEY>ID</KEY>
      <VALUE>57360</VALUE>
    </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Delete a Compliance Scan Schedule

/api/2.0/fo/schedule/scan/compliance/?action=delete&id=<id>

[POST]

Delete a scan schedule in the user’s account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) The ID of the scan schedule you want to delete.</td>
</tr>
<tr>
<td>echo_request=</td>
<td>(Optional) Specify 1 to echo the request’s input parameters (names and values) in the XML output. Otherwise parameters are not displayed in the output.</td>
</tr>
</tbody>
</table>

Sample - Delete compliance scan schedule

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/compliance/
?action=delete&id=57360"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2019-11-19T12:10:45Z</DATETIME>
    <TEXT>Schedule scan deleted successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>57360</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
Scan List Parameters

**Request Type**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required) A flag used to make a request for a scan list.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

**Filters**

Several parameters allow you to set filters to restrict the scan list output. When no filters are specified, the service returns all scans launched by all users within the past 30 days.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scan_ref={value}</td>
<td>(Optional) Show only a scan with a certain scan reference code. When unspecified, the scan list is not restricted to a certain scan. For a vulnerability scan, the format is: scan/987659876.19876. For a compliance scan the format is: compliance/98765456.12345. For a SCAP scan the format is: qscap/987659999.22222.</td>
</tr>
<tr>
<td>scan_id={value}</td>
<td>(Optional) Show only a scan with a certain compliance scan ID.</td>
</tr>
<tr>
<td>state={value}</td>
<td>(Optional) Show only one or more scan states. By default, the scan list is not restricted to certain states. A valid value is: Running, Paused, Canceled, Finished, Error, Queued (scan job is waiting to be distributed to scanner(s)), or Loading (scanner(s) are finished and scan results are being loaded onto the platform). Multiple values are comma separated.</td>
</tr>
<tr>
<td>processed={0</td>
<td>1}</td>
</tr>
<tr>
<td>type={value}</td>
<td>(Optional) Show only a certain scan type. By default, the scan list is not restricted to a certain scan type. A valid value is: On-Demand, Scheduled, or API.</td>
</tr>
<tr>
<td>target={value}</td>
<td>(Optional) Show only one or more target IP addresses. By default, the scan list includes all scans on all IP addresses. Multiple IP addresses and/or ranges may be entered. Multiple entries are comma separated. You may enter an IP address range using the hyphen (-) to separate the start and end IP address, as in: 10.10.10.1-10.10.10.2.</td>
</tr>
<tr>
<td>user_login={value}</td>
<td>(Optional) Show only a certain user login. The user login identifies a user who launched scans. By default, the scan list is not restricted to scans launched by a particular user. Enter the login name for a valid Qualys user account.</td>
</tr>
</tbody>
</table>
### Scans

#### Scan List Parameters

**launched_after_datetime=** `{date}` *(Optional)* Show only scans launched after a certain date and time (optional). The date/time is specified in `YYYY-MM-DD[THH:MM:SSZ]` format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”.

When `launched_after_datetime` and `launched_before_datetime` are unspecified, the service selects scans launched within the past 30 days.

A date/time in the future returns an empty scans list.

**launched_before_datetime=** `{date}` *(Optional)* Show only scans launched before a certain date and time (optional). The date/time is specified in `YYYY-MM-DD[THH:MM:SSZ]` format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”.

When `launched_after_datetime` and `launched_before_datetime` are unspecified, the service selects scans launched within the past 30 days.

A date/time in the future returns a list of all scans (not limited to scans launched within the past 30 days).

**scan_type=certview** *(Optional)* List CertView in VM scans only. This option will be supported when CertView GA is released and enabled for your account.

**scan_type=ec2certview** *(Optional)* List EC2 CertView VM scans only.

**client_id={value}** *(Optional)* Id assigned to the client (Consultant type subscriptions).

**client_name={value}** *(Optional)* Name of the client (Consultant type subscriptions).

Note: The `client_id` and `client_name` parameters are mutually exclusive and cannot be specified together in the same request.

---

### Show/Hide

These parameters specify whether certain information will be shown in the XML output.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`show_ags={0</td>
<td>1}`</td>
</tr>
<tr>
<td>`show_op={0</td>
<td>1}`</td>
</tr>
<tr>
<td>`show_status={0</td>
<td>1}`</td>
</tr>
<tr>
<td>`show_last={0</td>
<td>1}`</td>
</tr>
</tbody>
</table>
### Scan Parameters

Input parameters used to launch a VM or PC scan are below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={launch}</td>
<td>(Required) Specify &quot;launch&quot; to launch a new scan.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_title={value}</td>
<td>(Optional) The scan title. This can be a maximum of 2000 characters (ascii).</td>
</tr>
<tr>
<td>target_from={assets</td>
<td>tags}</td>
</tr>
<tr>
<td>ip={value}</td>
<td>(Optional) The IP addresses to be scanned. You may enter individual IP addresses and/or ranges. Multiple entries are comma separated. One of these parameters is required: ip, asset_groups or asset_group_ids.</td>
</tr>
<tr>
<td></td>
<td>ip is valid only when target_from=assets is specified.</td>
</tr>
<tr>
<td>asset_groups={value}</td>
<td>(Optional) The titles of asset groups containing the hosts to be scanned. Multiple titles are comma separated. One of these parameters is required: ip, asset_groups or asset_group_ids.</td>
</tr>
<tr>
<td></td>
<td>asset_groups is valid only when target_from=assets is specified.</td>
</tr>
<tr>
<td></td>
<td>These parameters are mutually exclusive and cannot be specified in the same request: asset_groups and asset_group_ids.</td>
</tr>
<tr>
<td>asset_group_ids={value}</td>
<td>(Optional) The IDs of asset groups containing the hosts to be scanned. Multiple IDs are comma separated. One of these parameters is required: ip, asset_groups or asset_group_ids.</td>
</tr>
<tr>
<td></td>
<td>asset_group_ids is valid only when target_from=assets is specified.</td>
</tr>
<tr>
<td></td>
<td>These parameters are mutually exclusive and cannot be specified in the same request: asset_groups and asset_group_ids.</td>
</tr>
</tbody>
</table>

** pci_only={0|1}  (Optional) Specify 1 to show only external PCI scans in the XML output. External PCI scans are vulnerability scans run with the option profile "Payment Card Industry (PCI) Options". When pci_only=1 is specified, the XML output will not include other types of scans run with other option profiles.  

** ignore_target={0|1}  (Optional) Specify 1 to hide target information from the scan list. Specify 0 to display the target information.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>exclude_ip_per_scan={value}</td>
<td>(Optional) The IP addresses to be excluded from the scan when the scan target is specified as IP addresses (not asset tags). You may enter individual IP addresses and/or ranges. Multiple entries are comma separated. exclude_ip_per_scan is valid only when target_from=assets is specified.</td>
</tr>
<tr>
<td>tag_include_selector= {all</td>
<td>any}</td>
</tr>
<tr>
<td>tag_exclude_selector= {all</td>
<td>any}</td>
</tr>
<tr>
<td>tag_set_by={id</td>
<td>name}</td>
</tr>
<tr>
<td>tag_set_include={value}</td>
<td>(Optional) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. tag_set_include is valid only when target_from=tags is specified.</td>
</tr>
<tr>
<td>tag_set_exclude={value}</td>
<td>(Optional) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. tag_set_exclude is valid only when target_from=tags is specified.</td>
</tr>
<tr>
<td>use_ip_nt_range_tags.include={0</td>
<td>1}</td>
</tr>
<tr>
<td>use_ip_nt_range_tags.exclude={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
### Scans

#### Scan Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`use_ip_nt_range_tags=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>iscanner_id={value}</code></td>
<td>(Optional) The IDs of the scanner appliances to be used. Multiple entries are comma separated. For an Express Lite user, Internal Scanning must be enabled in the user’s account. One of these parameters must be specified in a request: <code>iscanner_name</code>, <code>iscanner_id</code>, <code>default_scanner</code>, <code>scanners_in_ag</code>, <code>scanners_in_tagset</code>. When none of these are specified, External scanners are used. These parameters are mutually exclusive and cannot be specified in the same request: <code>iscanner_id</code> and <code>iscanner_name</code>.</td>
</tr>
<tr>
<td><code>iscanner_name={value}</code></td>
<td>(Optional) The friendly names of the scanner appliances to be used or “External” for external scanners. Multiple entries are comma separated. For an Express Lite user, Internal Scanning must be enabled in the user’s account. One of these parameters must be specified in a request for an internal scan: <code>iscanner_name</code>, <code>iscanner_id</code>, <code>default_scanner</code>, <code>scanners_in_ag</code>, <code>scanners_in_tagset</code>. When none of these are specified, External scanners are used. These parameters are mutually exclusive and cannot be specified in the same request: <code>iscanner_id</code> and <code>iscanner_name</code>.</td>
</tr>
<tr>
<td>`default_scanner=[0</td>
<td>1]`</td>
</tr>
</tbody>
</table>
### Scans

**Scan Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scanners_in_ag={0</td>
<td>1}</td>
</tr>
<tr>
<td>scanners_in_tagset={0</td>
<td>1}</td>
</tr>
<tr>
<td>scanners_in_network= {value}</td>
<td>(Optional) Specify 1 to distribute the scan to all scanner appliances in the network.</td>
</tr>
<tr>
<td>option_title={value}</td>
<td>(Optional) The title of the option profile to be used. One of these parameters must be specified in a request: option_title or option_id. These are mutually exclusive and cannot be specified in the same request.</td>
</tr>
<tr>
<td>option_id={value}</td>
<td>(Optional) The ID of the option profile to be used. One of these parameters must be specified in a request: option_title or option_id. These are mutually exclusive and cannot be specified in the same request.</td>
</tr>
<tr>
<td>priority={value}</td>
<td>(Optional for VM scans only) Specify a value of 0 - 9 to set a processing priority level for the scan. When not specified, a value of 0 (no priority) is used. Valid values are: 0 = No Priority (the default) 1 = Emergency 2 = Ultimate 3 = Critical 4 = Major 5 = High 6 = Standard 7 = Medium 8 = Minor 9 = Low</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>connector_name={value}</td>
<td>(Required for an EC2 scan) (VM scan only) The name of the EC2 connector for</td>
</tr>
<tr>
<td></td>
<td>the AWS integration you want to run the scan on.</td>
</tr>
<tr>
<td>ec2_endpoint={value}</td>
<td>(Required for an EC2 scan) The EC2 region code or the ID of the Virtual</td>
</tr>
<tr>
<td></td>
<td>Private Cloud (VPC) zone. Need help finding the region code? See the</td>
</tr>
<tr>
<td></td>
<td>following: <a href="http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-">http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-</a></td>
</tr>
<tr>
<td></td>
<td>regions-availability-zones.html#concepts-regions-availability-zones</td>
</tr>
<tr>
<td>ec2_instance_ids={value}</td>
<td>(Optional) (VM scan only) The ID of the EC2 instance on which you want to</td>
</tr>
<tr>
<td></td>
<td>launch the VM or compliance scan. Multiple ec2 instance ids are comma</td>
</tr>
<tr>
<td></td>
<td>separated. You can add up to maximum 10 instance IDs. When you launch an</td>
</tr>
<tr>
<td></td>
<td>EC2 scan and specify EC2 instance IDs as part of the scan target, we can</td>
</tr>
<tr>
<td></td>
<td>identify and skip any invalid instances and continue the scan on the valid</td>
</tr>
<tr>
<td></td>
<td>instances.</td>
</tr>
<tr>
<td>ip_network_id={value}</td>
<td>(Optional, and valid only when the Network Support feature is enabled for</td>
</tr>
<tr>
<td></td>
<td>the user’s account) The ID of a network used to filter the IPs/ranges</td>
</tr>
<tr>
<td></td>
<td>specified in the “ip” parameter. Set to a custom network ID (note this does</td>
</tr>
<tr>
<td></td>
<td>not filter IPs/ranges specified in “asset_groups” or “asset_group_ids”).</td>
</tr>
<tr>
<td></td>
<td>Or set to “0” (the default) for the Global Default Network - this is used</td>
</tr>
<tr>
<td></td>
<td>to scan hosts outside of your custom networks.</td>
</tr>
<tr>
<td>runtime_http_header={value}</td>
<td>(Optional) Set a custom value in order to drop defenses (such as logging,</td>
</tr>
<tr>
<td></td>
<td>IPs, etc) when an authorized scan is being run. The value you enter will be</td>
</tr>
<tr>
<td></td>
<td>used in the “Qualys-Scan:” header that will be set for many CGI and web</td>
</tr>
<tr>
<td></td>
<td>application fingerprinting checks. Some discovery and web server fingerprinting</td>
</tr>
<tr>
<td></td>
<td>checks will not use this header.</td>
</tr>
<tr>
<td>scan_type=certview</td>
<td>(Optional) (VM scan only) Launch a CertView type scan. This option will be</td>
</tr>
<tr>
<td></td>
<td>supported when CertView GA is released and enabled for your account.</td>
</tr>
<tr>
<td>fqdn={value}</td>
<td>(Optional) The target FQDN for a vulnerability scan. You must specify at</td>
</tr>
<tr>
<td></td>
<td>least one target i.e. IPs, asset groups or FQDNs. Multiple values are</td>
</tr>
<tr>
<td></td>
<td>comma separated.</td>
</tr>
<tr>
<td></td>
<td>- DNS Tracking must be enabled for the subscription. A Manager user can</td>
</tr>
<tr>
<td></td>
<td>enable this feature in the Qualys UI by going to Scans &gt; Setup &gt; DNS</td>
</tr>
<tr>
<td></td>
<td>Tracking and checking the &quot;Enable DNS Tracking for hosts&quot; option.</td>
</tr>
<tr>
<td></td>
<td>- You can specify FQDNs in combination with IPs and asset groups but not</td>
</tr>
<tr>
<td></td>
<td>with asset tags.</td>
</tr>
<tr>
<td>client_id={value}</td>
<td>(Optional) Id assigned to the client (Consultant type subscriptions).</td>
</tr>
</tbody>
</table>
# Cloud Perimeter Scan Parameters

The input parameters for creating or updating a Cloud Perimeter scan are below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client_name={value}</td>
<td>(Optional) Name of the client (Consultant type subscriptions). Note: The client_id and client_name parameters are mutually exclusive and cannot be specified together in the same request.</td>
</tr>
</tbody>
</table>
| include_agent_targets={0|1} | (Optional) Specify 1 when your scan target includes agent hosts. This lets you scan private IPs where agents are installed when these IPs are not in your VM/PC license. Supported capabilities  
  - This parameter is supported for internal scans using scanner appliance(s). This option is not supported for scans using External scanners.  
  - This parameter is supported when launching on demand scans only. It is not supported for scheduled scans.  
  Parameter iscanner_id or iscanner_name must be specified in the same request. |

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={create</td>
<td>update}</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required and only applicable for Update request) The ID of the scan schedule you want to update.</td>
</tr>
<tr>
<td>module={vm</td>
<td>pc}</td>
</tr>
</tbody>
</table>
| cloud_provider={value}     | (Optional) Specify "azure" for an Azure scan. Specify "aws" for an AWS EC2 scan. The cloud_provider value cannot be changed during an update request.  
  When cloud_provider=azure, the following parameters cannot be specified in the same request: platform_type, region_code, vpc_id, include_micro_nano_instances, include_lb_from_connector. These parameters only apply when cloud_provider=aws is specified. |
| cloud_service={value}      | (Optional) Specify "vm" (Azure virtual machine) for an Azure scan. Specify "ec2" for an AWS EC2 scan. The cloud_service value cannot be changed during an update request. |
### Scans

#### Cloud Perimeter Scan Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connector_name={value}</td>
<td>(Optional) The name of the connector to be used. We will check if the specified connector_name exists for your Qualys subscription. If not, then API request returns an error message &quot;Invalid connector_name provided&quot;. One of these parameters must be specified in the request: connector_name or connector_uuid. These are mutually exclusive and cannot be specified in the same request.</td>
</tr>
<tr>
<td>connector_uuid={value}</td>
<td>(Optional) The ID of the connector to be used. We will check if the specified connector_uuid exists for your Qualys subscription. If not, then API request returns an error message &quot;Invalid connector_uuid provided&quot;. One of these parameters must be specified in the request: connector_name or connector_uuid. These are mutually exclusive and cannot be specified in the same request.</td>
</tr>
<tr>
<td>scan_title={value}</td>
<td>(Optional) The scan title. When not specified the default scan title is &quot;AWS EC2 Perimeter Scan &lt;date&gt;&quot;</td>
</tr>
<tr>
<td>active={0</td>
<td>1}</td>
</tr>
<tr>
<td>option_title={value}</td>
<td>(Optional) The title of the option profile to be used. One of these parameters must be specified in the request: option_title or option_id. These are mutually exclusive and cannot be specified in the same request.</td>
</tr>
<tr>
<td>option_id={value}</td>
<td>(Optional) The ID of the option profile to be used. One of these parameters must be specified in a request: option_title or option_id. These are mutually exclusive and cannot be specified in the same request.</td>
</tr>
<tr>
<td>priority={value}</td>
<td>(Optional) Specify a value of 0 - 9 to set a processing priority level for the scan. When not specified, a value of 0 (no priority) is used. Valid values are: 0 = No Priority (the default) 1 = Emergency 2 = Ultimate 3 = Critical 4 = Major 5 = High 6 = Standard 7 = Medium 8 = Minor 9 = Low</td>
</tr>
<tr>
<td>iscanner_id={value}</td>
<td>(Optional, only valid when your account is configured to allow internal scanners) The IDs of the scanner appliances to be used. Specify &quot;0&quot; for external scanners. Multiple entries are comma separated. These parameters cannot be specified in the same request: iscanner_id and iscanner_name.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>iscanner_name={value}</td>
<td>(Optional, only valid when your account is configured to allow internal scanners) The friendly names of the scanner appliances to be used or &quot;External&quot; for external scanners. Multiple entries are comma separated. These parameters cannot be specified in the same request: iscanner_id and iscanner_name.</td>
</tr>
<tr>
<td>platform_type={value}</td>
<td>(Optional) The platform type. Valid values are: classic, vpc_peered or selected_vpc.</td>
</tr>
<tr>
<td>region_code={value}</td>
<td>(Optional) The EC2 region code. Valid values are: ap-northeast-1, ap-southeast-1, ap-southeast-2, ap-east-1, eu-west-1, eu-north-1, asa-east-1, us-east-1, us-west-1, us-west-2, me-south-1, eu-south-1, and af-south-1. One of these parameters must be specified in the request: region_code or vpc_id. These are mutually exclusive and cannot be specified in the same request.</td>
</tr>
<tr>
<td>vpc_id={value}</td>
<td>(Optional) The ID of the Virtual Private Cloud (VPC) zone. The ID value must start with vpc-*. We will check if the specified vpc_id exists for the selected connector. One of these parameters must be specified in the request: region_code or vpc_id. These are mutually exclusive and cannot be specified in the same request.</td>
</tr>
<tr>
<td>include_micro_nano_instances={0</td>
<td>1}</td>
</tr>
<tr>
<td>tag_include_selector=</td>
<td>(Optional) Select &quot;any&quot; (the default) to include hosts that match at least one of the selected tags. Select &quot;all&quot; to include hosts that match all of the selected tags.</td>
</tr>
<tr>
<td>tag_exclude_selector=</td>
<td>(Optional) Select &quot;any&quot; (the default) to exclude hosts that match at least one of the selected tags. Select &quot;all&quot; to exclude hosts that match all of the selected tags.</td>
</tr>
<tr>
<td>tag_set_by=</td>
<td>(Optional) Specify &quot;id&quot; (the default) to select a tag set by providing tag IDs. Specify &quot;name&quot; to select a tag set by providing tag names. We will check if the tag ids or tag names are valid.</td>
</tr>
</tbody>
</table>
### Cloud Perimeter Scan Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag_set_include={value}</td>
<td>(Optional) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>tag_set_exclude={value}</td>
<td>(Optional) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>include_lb_from_connector</td>
<td>(Optional) Specify 1 to include public load balancers from the selected connector in the scan job. By default, this parameter value is set to 0.</td>
</tr>
<tr>
<td>elb_dns={value}</td>
<td>(Optional) One or more load balancer DNS names to include in the scan job. Multiple values are comma-separated.</td>
</tr>
<tr>
<td>schedule={value}</td>
<td>(Required for Create request) Specify &quot;now&quot; to schedule the scan job for now. Specify &quot;recurring&quot; to schedule the scan job to start at a later time or on a recurring basis. See Scheduling Parameters in the next section.</td>
</tr>
</tbody>
</table>
# Scan Schedule Parameters

## Scan Schedule - Occurrence

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>occurrence=daily</td>
<td>Required for a daily scan.</td>
</tr>
<tr>
<td>frequency_days={value}</td>
<td>Required for a daily scan. The scan will run every N number of days. Value is an integer from 1 to 365.</td>
</tr>
<tr>
<td>occurrence=weekly</td>
<td>Required for a weekly scan.</td>
</tr>
<tr>
<td>frequency_weeks={value}</td>
<td>Required for a weekly scan. The scan will run every N number of weeks. Value is an integer from 1 to 52.</td>
</tr>
<tr>
<td>weekdays={value}</td>
<td>Required for a weekly scan. The scan will run on the one or more weekdays. Value is one or more days: sunday, monday, tuesday, wednesday, thursday, friday, saturday. Multiple days are comma separated.</td>
</tr>
<tr>
<td>occurrence=monthly</td>
<td>Required for a monthly scan.</td>
</tr>
<tr>
<td>frequency_months={value}</td>
<td>Required for a monthly scan. The scan will run every N number of months. Value is an integer from 1 to 12.</td>
</tr>
<tr>
<td>day_of_month={value}</td>
<td>Required for monthly scan - Nth day of the month. The scan will run on the Nth day of the month. Value is an integer from 1 to 31.</td>
</tr>
<tr>
<td>day_of_week={value}</td>
<td>Required for monthly scan - day in Nth week. The scan will run on this day of the week. Value is and integer from 0 to 6, where 0 is Sunday and 2 is Tuesday.</td>
</tr>
<tr>
<td>week_of_month={value}</td>
<td>Required for monthly scan - day in Nth week. The scan will run on this week of the month. Value is one of: first, second, third, fourth, last.</td>
</tr>
</tbody>
</table>

## Scan Schedule - Start Time

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_date={mm/dd/yyyy}</td>
<td>(Optional) By default the start date is the date when the schedule is created. You can define another start date in mm/dd/yyyy format.</td>
</tr>
<tr>
<td>start_hour={hour}</td>
<td>(Required) The hour when a scan will start. The hour is an integer from 0 to 23, where 0 represents 12 AM, 7 represents 7 AM, and 22 represents 10 PM.</td>
</tr>
<tr>
<td>start_minute={minute}</td>
<td>(Required) The minute when a scan will start. A valid value is an integer from 0 to 59.</td>
</tr>
<tr>
<td>time_zone_code={value}</td>
<td>(Required) The time zone code for starting a scan, in upper case. For example, the time zone code for US California is US-CA. Valid codes are returned by the Time Zone Code API (/msp/time_zone_code_list.php).</td>
</tr>
<tr>
<td>observe_dst={yes</td>
<td>no}</td>
</tr>
</tbody>
</table>
### Scan Schedule Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recurrence={value}</td>
<td>(Optional) The number of times the scan will be run before it is deactivated. For example, if you set recurrence=2, the scan schedule will be deactivated after it runs 2 times. By default no value is set. A valid value is an integer from 1 to 99.</td>
</tr>
<tr>
<td>end_after={value}</td>
<td>(Optional) End a scan after some number of hours. A valid value is from 0 to 119.</td>
</tr>
<tr>
<td>end_after_mins={value}</td>
<td>(Optional) End a scan after some number of minutes. A valid value is an integer from 0 to 59. Must be specified with end_after. For example, to end the scan after 2 hours and 30 minutes, you would specify end_after=2 and end_after_mins=30. When end_after is set to 0, the minimum value for end_after_mins is 15.</td>
</tr>
<tr>
<td>pause_after_hours={value}</td>
<td>(Optional) Pause a scan after some number of hours if the scan has not finished by then. A valid value is an integer from 0 to 119.</td>
</tr>
<tr>
<td>pause_after_mins={value}</td>
<td>(Optional) Pause a scan after some number of minutes if the scan has not finished by then. A valid value is an integer from 0-59. Must be specified with pause_after_hours. For example, to pause the scan after 2 hours and 30 minutes, you would specify pause_after_hours=2 and pause_after_mins=30. When pause_after_hours is set to 0, the minimum value for pause_after_mins is 15.</td>
</tr>
<tr>
<td>resume_in_days={value}</td>
<td>(Optional) Resume a paused scan in some number of days. A valid value is an integer from 0 to 9 or Manually.</td>
</tr>
<tr>
<td>resume_in_hours={value}</td>
<td>(Optional) Resume a paused scan in some number of hours. A valid value is an integer from 0-23. Must be specified with pause_after_hours and resume_in_days. For example, to resume your scan in 5 hours, specify resume_in_days=0 and resume_in_hours=5. To resume your scan in 1 day and 12 hours, specify resume_in_days=1 and resume_in_hours=12. Note - The value you set for pause will determine the minimum value for resume. For example, if you set the scan to pause after 1 hour then you can set it to resume in 2 or more hours. If you set the scan to pause between 1-2 hours (from 1hr, 1min to 1 hr, 59min) then you can set it to resume in 3 hours or more.</td>
</tr>
<tr>
<td>set_start_time={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
### Scan Schedule - Notifications

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>before_notify=[0</td>
<td>1]</td>
</tr>
<tr>
<td>before_notify_unit={value}</td>
<td>(Optional) Specify the time unit for when to send the before scan notification. Possible values are: days, hours, minutes. This parameter is required when before_notify=1. Not valid when before_notify=0.</td>
</tr>
<tr>
<td>before_notify_time={value}</td>
<td>(Optional) Indicates the number of days, hours or minutes before the scan starts the notification will be sent. For days, enter a value of 1-31. For hours, enter a value of 1-24. For minutes, enter a value of 5-120. This parameter is required when before_notify=1. Not valid when before_notify=0.</td>
</tr>
<tr>
<td>before_notify_message=</td>
<td>(Optional) Specify a custom message to add to the before scan notification. The notification will always include certain details like the scan title, owner, option profile and start time. Include up to 4000 characters, no HTML tags. For update requests:</td>
</tr>
<tr>
<td></td>
<td>- When not specified we keep the previous setting.</td>
</tr>
<tr>
<td></td>
<td>- Specify an empty string to delete the last saved message.</td>
</tr>
<tr>
<td></td>
<td>This parameter is only valid when before_notify=1.</td>
</tr>
<tr>
<td>after_notify=[0</td>
<td>1]</td>
</tr>
<tr>
<td>after_notify_message=</td>
<td>(Optional) Specify a custom message to add to the after scan notification. When not specified during a create request, no notification message is saved. Include up to 4000 characters, no HTML tags. For update requests:</td>
</tr>
<tr>
<td></td>
<td>- When not specified we keep the previous setting.</td>
</tr>
<tr>
<td></td>
<td>- Specify an empty string to delete the last saved message.</td>
</tr>
<tr>
<td></td>
<td>- If both notifications are disabled (before_notify=0 and after_notify=0) we will delete the after notify message.</td>
</tr>
<tr>
<td></td>
<td>This parameter is only valid when after_notify=1.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| `recipient_group_ids={value}`    | (Optional) The notification recipients in the form of one or more valid distribution group IDs. When not specified during a create request, only the task owner will be notified. For update requests:  
  - When not specified we keep the previous setting.  
  - Specify an empty string to delete the list of IDs.  
  - If both notifications are disabled (before_notify=0 and after_notify=0) we will delete the list of IDs.  
  This parameter is only valid when before_notify=1 or after_notify=1 is specified in the same request. |
| `delay_notify={0|1}`             | (Optional) Specify to send a notification if a scheduled scan is delayed.                                                                    |
| `delay_notify_message={value}`   | (Optional) Specify a message to send notification for a delayed scheduled scan. If a message is not specified or if the delay_notify=1, the following default message is shown: “The Qualys scan launch has been delayed and will be tried again.” This parameter is only valid when delay_notify=1. |
| `skipped_notify={0|1}`           | (Optional) Specify to send a notification if a scheduled scan is skipped.                                                                     |
| `skipped_notify_message={value}` | (Optional) Specify a message to send notification for a skipped scheduled scan. If a message is not specified or if the skipped_notify=1, the following default message is shown: “The Qualys scan launch has been skipped.” This parameter is only valid when skipped_notify=1. |
| `deactivate_notify={0|1}`        | (Optional) Specify to send a notification if a scheduled scan is deactivated.                                                                |
| `deactivate_notify_message={value}` | (Optional) Specify a message to send notification for a deactivated scheduled scan. If a message is not specified or if the deactivate_notify=1, the following default message is shown: “The Qualys scan has been deactivated by the service.” This parameter is only valid when deactivate_notify=1. |

### Scan Schedule - Consultant type subscriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>client_id={value}</code></td>
<td>(Optional) Id assigned to the client (Consultant type subscriptions).</td>
</tr>
<tr>
<td><code>client_name={value}</code></td>
<td>(Optional) Name of the client (Consultant type subscriptions). Note: The client_id and client_name parameters are mutually exclusive and cannot be specified together in the same request.</td>
</tr>
</tbody>
</table>
VM Scan Statistics

/api/2.0/fo/scan/stats/?action=list

[GET] [POST]

List details about vulnerability scans and assets that are waiting to be processed.

Permissions - Manager role is required.

You’ll see these sections in the XML output:

UNPROCESSED SCANS - The total number of scans that are not processed, including scans that are queued, running, loading, finished, etc.

VM RECRYPT BACKLOG - The total number of assets across your finished scans that are waiting to be processed.

VM RECRYPT BACKLOG BY SCAN - Scan details for vulnerability scans that are waiting to be processed. For each scan, you’ll see the scan ID, scan title, scan status, processing priority and number of hosts that the scan finished but not processed.

VM RECRYPT BACKLOG BY TASK - Processing task details for vulnerability scans that are waiting to be processed. For each task, you’ll see the same scan details as VM RECRYPT BACKLOG BY SCAN plus additional information like the total hosts alive for the scan, the number of hosts from the scan that have been processed, the number of hosts waiting to be processed, the scan start date, the task type and task status.

Sample - List VM statistics

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl"
"https://qualysapi.qualys.com/api/2.0/fo/scan/stats/?action=list"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TASK_PROCESSING SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/stats/vm_recrypt_results.dtd">

<TASK_PROCESSING>
  <UNPROCESSED_SCANS><![CDATA[366]]></UNPROCESSED_SCANS>
  <VM_RECRYPT_BACKLOG><![CDATA[116]]></VM_RECRYPT_BACKLOG>
  <VM_RECRYPT_BACKLOG_BY_SCAN>
    <SCAN>
      <ID><![CDATA[189275]]></ID>
      <TITLE><![CDATA[API_V2_IP_Scan_1511513769]]></TITLE>
      <STATUS><![CDATA[Loading]]></STATUS>
    </SCAN>
  </VM_RECRYPT_BACKLOG_BY_SCAN>
</TASK_PROCESSING>
<ID><![CDATA[215356]]></ID>
<TITLE><![CDATA[API_V2_AG_Scan_1515742250]]></TITLE>
<STATUS><![CDATA[Running]]></STATUS>

<PROCESSING_PRIORITY><![CDATA[None]]></PROCESSING_PRIORITY>
<NBHOST><![CDATA[]]></NBHOST>
<TO_PROCESS><![CDATA[0]]></TO_PROCESS>
<PROCESSED><![CDATA[0]]></PROCESSED>
<SCAN_DATE><![CDATA[2018-01-12T07:30:42Z]]></SCAN_DATE>
<SCAN_UPDATED_DATE><![CDATA[2018-01-12T08:01:10Z]]></SCAN_UPDATED_DATE>
<TASK_TYPE><![CDATA[VM Scan Processing]]></TASK_TYPE>
<TASK_STATUS><![CDATA[Queued]]></TASK_STATUS>
<TASK_UPDATED_DATE><![CDATA[2018-01-12T08:17:11Z]]></TASK_UPDATED_DATE>

</SCAN>

<ID><![CDATA[215357]]></ID>
<TITLE><![CDATA[API_V2_AG_Scan_1515742265]]></TITLE>
<STATUS><![CDATA[Loading]]></STATUS>

<PROCESSING_PRIORITY><![CDATA[None]]></PROCESSING_PRIORITY>
<NBHOST><![CDATA[]]></NBHOST>
<TO_PROCESS><![CDATA[0]]></TO_PROCESS>
<PROCESSED><![CDATA[0]]></PROCESSED>
<SCAN_DATE><![CDATA[2018-01-12T07:30:58Z]]></SCAN_DATE>
<SCAN_UPDATED_DATE><![CDATA[2018-01-12T08:14:45Z]]></SCAN_UPDATED_DATE>
<TASK_TYPE><![CDATA[VM Scan Processing]]></TASK_TYPE>
<TASK_STATUS><![CDATA[Queued]]></TASK_STATUS>
<TASK_UPDATED_DATE><![CDATA[2018-01-12T08:17:11Z]]></TASK_UPDATED_DATE>

</SCAN>

...</VM_RECRYPT_BACKLOG_BY_TASK>
</TASK_PROCESSING>

DTD
<platform API server>/api/2.0/fo/scan/stats/vm_recrypt_results.dtd
VM Scan Summary

/api/2.0/fo/scan/vm/summary/

[GET]  [POST]

This API can be used as an alternative to the original Scan Summary API. It’s easier to use, has more filter options and enhanced output content.

This API helps you to identify hosts that were scanned or not scanned and why. You can choose to get a scan summary for a particular scan by specifying the scan reference ID or for all scans launched since a certain date/time or within a date range.

This API will return details for all scans. Note, however, that the output will not include the <SCAN_RESULTS> block if the scan did not return results for some reason.

Permissions
Manager role is required.

Input Parameters
The following input parameters are supported.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required) The list action.</td>
</tr>
<tr>
<td>output_format=xml</td>
<td>(Optional) The only supported output format at this time is XML.</td>
</tr>
<tr>
<td>scan_reference={value}</td>
<td>(Optional) Specifies a unique scan reference ID. Use this option to include scan summary information for a single scan only. For VM scans, the scan reference has the format scan/987654321.98765.</td>
</tr>
<tr>
<td></td>
<td>One of these parameters must be specified in the request: scan_datetime_since or scan_reference. You cannot specify scan_reference in the same request as scan_datetime_since and scan_datetime_until.</td>
</tr>
<tr>
<td>scan_datetime_since={value}</td>
<td>(Optional) Include scans started since a certain date. The date must be less than or equal to today’s date. Specify the date in GMT timezone in RFC 3339 format: yyyy-mm-ddThh-mm-ssZ. Example: 2020-10-01T09:30:48Z</td>
</tr>
<tr>
<td></td>
<td>One of these parameters must be specified in the request: scan_datetime_since or scan_reference. You cannot specify scan_datetime_since in the same request as scan_reference.</td>
</tr>
<tr>
<td>scan_datetime_until={value}</td>
<td>(Optional) Include scans started up to a certain date. The date must be more than or equal to scan_datetime_since, and less than or equal to today’s date. Specify the date in GMT timezone in RFC 3339 format: yyyy-mm-ddThh-mm-ssZ. Example: 2020-10-01T09:30:48Z</td>
</tr>
<tr>
<td></td>
<td>The parameter scan_datetime_until can only be specified when scan_datetime_since is also specified. You cannot specify scan_datetime_until in the same request as scan_reference.</td>
</tr>
</tbody>
</table>
VM Scan Summary

The following host summary categories may be included in the scan summary output:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_scan_input={0</td>
<td>1}</td>
</tr>
<tr>
<td>include_scan_details={0</td>
<td>1}</td>
</tr>
<tr>
<td>include_hosts_summary={0</td>
<td>1}</td>
</tr>
<tr>
<td>include_detections_summary={0</td>
<td>1}</td>
</tr>
<tr>
<td>include_hosts_summary_categories={value}</td>
<td>(Optional) When unspecified, all categories are included in the XML output. To filter the categories, provide a comma-separated list of the categories to include in the output. Possible values are: scanned, excluded, cancelled, unresolved, duplicate, not_vulnerable, dead, aborted, blocked, failed_slice, exceeded_scan_duration.</td>
</tr>
</tbody>
</table>

See Host Summary Categories below for more information on each category. Each category appears a block inside <SCANRESULTS> <HOSTS>. If a category is filtered out, the respective category block does not appear in the output.

**Host Summary Categories**

The following host summary categories may be included in the scan summary output:

- **Scanned** - The hosts were scanned successfully.
- **Excluded** - The hosts were excluded. Hosts may be excluded on a per scan basis (by the user launching or scheduling the scan) or globally for all scans. Managers and Unit Managers have privileges to edit the global excluded hosts list for the subscription.
- **Cancelled** - Hosts were not scanned because the scan was cancelled. Scans may be cancelled by a user, by an administrator or automatically by the service as specified in scheduled scan settings.
Unresolved - Hosts were scanned but they could not be reported because the NetBIOS or DNS hostname, whichever tracking method is specified for each host, could not be resolved.

Duplicate - The hosts were duplicated within a single segment/slice of the scan job. For example, two different hostnames resolving to the same IP with tracking by IP.

Not Vulnerable - Hosts were found to be not vulnerable during host discovery without having to run a full scan. This could happen for example if the list of QIDs to be scanned are limited to certain ports and those ports are found to be closed.

Dead - The hosts were not “alive” at the time of the scan, meaning that they did not respond to probes sent by the scanning engine, and the option to Scan Dead Hosts was not enabled.

Aborted - The scan was abruptly discontinued. This is a rare occurrence that may be caused for different reasons. For example, it’s possible that a connection timed out or there were connection errors on a particular port or the scan time elapsed.

Blocked - Hosts were blocked from scanning for some reason. For example, user provided blacklisted IPs to scan and after the scan was launched it was blocked due to improper configuration.

Failed Slice Hosts - The scan failed for these hosts.

Exceeded Scan Duration - Applicable when the Maximum Scan Duration per Asset feature is enabled and a maximum scan duration is specified in the option profile used for the scan. This setting determines how long a scan can run on a single asset. The scan on these hosts exceeded the scan duration allowed so the scan on these hosts was aborted.

**Sample 1 - Get scan summary by scan reference**

In this sample, the scan reference ID is included as part of the request.

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" "http://qualysapi.qualys.com/api/2.0/fo/scan/vm/summary/?action=list&include_scan_input=1&include_hosts_summary=1&include_detections_summary=1&scan_reference=scan/9876543210.12345"
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCAN_SUMMARY_OUTPUT SYSTEM "http://qualysapi.qualys.com/api/2.0/fo/scan/vm/summary/output.dtd">
<SCAN_SUMMARY_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-09-15T09:09:36Z</DATETIME>
    <SCAN_SUMMARY_LIST>
      <SCAN_SUMMARY>
        <SCAN_REFERENCE>scan/9876543210.12345</SCAN_REFERENCE>
        <SCAN_INPUT>
          <TITLE>CustomAgScan</TITLE>
          <USER>
            <USERNAME>qualys_joe</USERNAME>
          </USER>
        </SCAN_INPUT>
      </SCAN_SUMMARY>
    </SCAN_SUMMARY_LIST>
  </RESPONSE>
</SCAN_SUMMARY_OUTPUT>
```
<USER>
</SCHEDULED>
</SCAN_DATETIME> 2020-04-01 06:55:55</SCAN_DATETIME>
</NETWORK>
  <ID>63010</ID>
  <NAME>Custom Network</NAME>
</NETWORK>
  <OPTION_PROFILE>
    <ID>86395</ID>
    <NAME>Initial Options</NAME>
  </OPTION_PROFILE>
</TARGETS>
  <IP_LIST>
    <COUNT>256</COUNT>
    <IP_DATA>
      <RANGES>
        <RANGE>11.1.1.0-11.1.1.255</RANGE>
      </RANGES>
    </IP_DATA>
  </IP_LIST>
  <DNS_LIST>
    <COUNT>3</COUNT>
    <DNS_DATA>
      <DNS_CSV>sample2.com, sample3.com, sample1.com</DNS_CSV>
    </DNS_DATA>
  </DNS_LIST>
  <NETBIOS_LIST>
    <COUNT>3</COUNT>
    <NETBIOS_DATA>
      <NETBIOS_CSV>NB1.COM, NB3.COM, NB2.COM</NETBIOS_CSV>
    </NETBIOS_DATA>
  </NETBIOS_LIST>
  <ASSET_GROUP_LIST>
    <COUNT>1</COUNT>
    <ASSET_GROUP_DATA>
      <ASSET_GROUP>
        <ID>206216</ID>
        <NAME>Custom Network Asset Group</NAME>
      </ASSET_GROUP>
    </ASSET_GROUP_DATA>
  </ASSET_GROUP_LIST>
</TARGETS>
</SCAN_INPUT>
</SCANDETAILS>
  <STATUS>ERROR</STATUS>
  <LAUNCH_DATETIME>2020-04-01 06:55:55</LAUNCH_DATETIME>
  <DURATION>1261</DURATION>
</SCANDETAILS>
<SCANRESULTS>
  <HOSTS>
    <COUNT>262</COUNT>
    <HOSTS_DATA>
<SCANNED>
  <IP_LIST>
    <COUNT>9</COUNT>
    <IP_DATA>
      <RANGES>
        <RANGE>43.56.78.111-43.56.78.119</RANGE>
      </RANGES>
    </IP_DATA>
  </IP_LIST>
  </SCANNED>
  <FAILED_SLICE_HOSTS>
    <IPV4_LIST>
      <COUNT>8</COUNT>
      <IPV4_DATA>
        <IPV4_CSV>10.10.10.1,10.20.10.10-10.20.10.13,10.10.10.3,10.20.10.7,10.10.10.8,10.20.10.11</IPV4_CSV>
      </IPV4_DATA>
    </IPV4_LIST>
    <IPV6_LIST>
      <COUNT>2</COUNT>
      <IPV6_DATA>
        <IPV6_CSV>::ff01,::ff02,::ff02</IPV6_CSV>
      </IPV6_DATA>
    </IPV6_LIST>
    <DNS_LIST>
      <COUNT>4</COUNT>
      <DNS_DATA>
        <DNS_CSV>sample4.com, sample5.com, sample6.com, sample7.com</DNS_CSV>
      </DNS_DATA>
    </DNS_LIST>
    <NETBIOS_LIST>
      <COUNT>4</COUNT>
      <NETBIOS_DATA>
        <NETBIOS_CSV>SATEELITE,WIN4KB, KRWSDG</NETBIOS_CSV>
      </NETBIOS_DATA>
    </NETBIOS_LIST>
  </FAILED_SLICE_HOSTS>
  </HOSTS_DATA>
  <DETECTIONS>
    <IG>
      <TOTAL_COUNT>7216</TOTAL_COUNT>
      <COUNT_BY_SEVERITY>
        <SEVERITY_1>4467</SEVERITY_1>
        <SEVERITY_2>2232</SEVERITY_2>
        <SEVERITY_3>517</SEVERITY_3>
        <SEVERITY_4>0</SEVERITY_4>
        <SEVERITY_5>0</SEVERITY_5>
      </COUNT_BY_SEVERITY>
    </IG>
  </DETECTIONS>
</VM Scan Summary>
Scans
VM Scan Summary

<VULN>
  <CONFIRMED>
    <TOTAL_COUNT>8054</TOTAL_COUNT>
    <COUNT_BY_SEVERITY>
      <SEVERITY_1>238</SEVERITY_1>
      <SEVERITY_2>985</SEVERITY_2>
      <SEVERITY_3>2124</SEVERITY_3>
      <SEVERITY_4>2546</SEVERITY_4>
      <SEVERITY_5>2161</SEVERITY_5>
    </COUNT_BY_SEVERITY>
  </CONFIRMED>
  <POTENTIAL>
    <TOTAL_COUNT>1497</TOTAL_COUNT>
    <COUNT_BY_SEVERITY>
      <SEVERITY_1>17</SEVERITY_1>
      <SEVERITY_2>420</SEVERITY_2>
      <SEVERITY_3>579</SEVERITY_3>
      <SEVERITY_4>304</SEVERITY_4>
      <SEVERITY_5>177</SEVERITY_5>
    </COUNT_BY_SEVERITY>
  </POTENTIAL>
</VULN>

Sample 2 - Get scan summary by scan date
In this sample, all scans within the date range will be returned.

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/scan/vm/summary/?action=list&output_format=xml&scan_datetime_since=2020-04-06T02:30:00Z&scan_datetime_until=2020-04-06T02:30:00Z"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCAN_SUMMARY_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/vm/summary/output.dtd">
<SCAN_SUMMARY_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-09-22T05:02:40Z</DATETIME>
    <SCAN_SUMMARY_LIST>
      <SCAN_SUMMARY>
        <SCAN_REFERENCE>scan/1234567890.12345</SCAN_REFERENCE>
        <SCAN_INPUT>
          <TITLE>My-Scan</TITLE>
          <USER>

101
<USERNAME>qualys_joe</USERNAME>
</USER>
<SCHEDULED>0</SCHEDULED>
<SCAN_DATETIME>2020-04-06 07:17:45</SCAN_DATETIME>
<N畦>0</ID>
<NAME>Global Default Network</NAME>
</NETWORK>
<OPTION_PROFILE>
<ID>2171</ID>
<NAME>Initial Options</NAME>
</OPTION_PROFILE>
<TARGETS>
<IP_LIST>
<COUNT>3</COUNT>
<IP_DATA>
<RANGES>
  <RANGE>10.10.30.10-10.10.30.12</RANGE>
</RANGES>
</IP_DATA>
</IP_LIST>
</TARGETS>
</SCAN_INPUT>
<SCAN_DETAILS>
<STATUS>FINISHED</STATUS>
<LAUNCH_DATETIME>2020-04-06 07:17:45</LAUNCH_DATETIME>
<DURATION>21656</DURATION>
</SCAN_DETAILS>
<SCAN_RESULTS>
<HOSTS>
<COUNT>3</COUNT>
<HOSTS_DATA>
<SCANNED>
<IP_LIST>
<COUNT>2</COUNT>
<IP_DATA>
<RANGES>
  <RANGE>10.10.30.10-10.10.30.11</RANGE>
</RANGES>
</IP_DATA>
</IP_LIST>
</SCANNED>
<DEAD>
<IP_LIST>
<COUNT>1</COUNT>
<IP_DATA>
<IP_CSV>10.10.30.12</IP_CSV>
</IP_DATA>
</IP_LIST>
</DEAD>
<FAILED_SLICE_HOSTS>
<IPV4_LIST>
<COUNT>8</COUNT>
</IPV4_LIST>
</FAILED_SLICE_HOSTS>
</SCAN_RESULTS>
<IPV4_CSV>10.10.10.1, 10.20.10.10-10.20.10.13, 10.10.10.3, 10.20.10.7, 10.10.10.8, 10.20.10.11</IPV4_CSV>
</IPV4_DATA>
<IPV6_LIST>
  <COUNT>2</COUNT>
  <IPV6_DATA>
    <IPV6_CSV>::ff01, ::ff02, ::ff02</IPV6_CSV>
  </IPV6_DATA>
</IPV6_LIST>
<DNS_LIST>
  <COUNT>4</COUNT>
  <DNS_DATA>
    <DNS_CSV>sample1.com, sample2.com, sample3.com, sample4.com</DNS_CSV>
  </DNS_DATA>
</DNS_LIST>
<NETBIOS_LIST>
  <COUNT>4</COUNT>
  <NETBIOS_DATA>
    <NETBIOS_CSV>SAMPLE1, SAMPLE2, SAMPLE3, SAMPLE4</NETBIOS_CSV>
  </NETBIOS_DATA>
</NETBIOS_LIST>
</FAILED_SLICE_HOSTS>
</HOSTS_DATA>
</HOSTS>
<DETECTIONS>
  <IG>
    <TOTAL_COUNT>77</TOTAL_COUNT>
    <COUNT_BY_SEVERITY>
      <SEVERITY_1>52</SEVERITY_1>
      <SEVERITY_2>12</SEVERITY_2>
      <SEVERITY_3>5</SEVERITY_3>
      <SEVERITY_4>2</SEVERITY_4>
      <SEVERITY_5>6</SEVERITY_5>
    </COUNT BY SEVERITY>
  </IG>
  <VULN>
    <CONFIRMED>
      <TOTAL_COUNT>17</TOTAL_COUNT>
      <COUNT BY SEVERITY>
        <SEVERITY_1>0</SEVERITY_1>
        <SEVERITY_2>3</SEVERITY_2>
        <SEVERITY_3>10</SEVERITY_3>
        <SEVERITY_4>0</SEVERITY_4>
        <SEVERITY_5>4</SEVERITY_5>
      </COUNT BY SEVERITY>
    </CONFIRMED>
    <POTENTIAL>
      <TOTAL_COUNT>18</TOTAL_COUNT>
      <COUNT BY SEVERITY>
        <SEVERITY_1>2</SEVERITY_1>
        <SEVERITY_2>4</SEVERITY_2>
        <SEVERITY_3>10</SEVERITY_3>
  </POTENTIAL>
</DETECTIONS>
<SEVERITY_4>1</SEVERITY_4>
<SEVERITY_5>1</SEVERITY_5>
</COUNT_BY_SEVERITY>
</POTENTIAL>
</VULN>
</DETECTIONS>
</SCAN_RESULTS>
</SCAN_SUMMARY>

---

**Sample 3 - Filter list of categories in output**

In this sample, only the Cancelled category is included.

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/scan/vm/summary/?action=list&output_format=xml&scan_reference=scan/1234567890.12345&include_hosts_summary_categories=cancelled"
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCAN_SUMMARY_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/vm/summary/output.dtd">
<SCAN_SUMMARY_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-09-22T05:07:05Z</DATETIME>
    <SCAN_SUMMARY_LIST>
      <SCAN_SUMMARY>
        <SCAN_REFERENCE>scan/1234567890.12345</SCAN_REFERENCE>
        <SCAN_INPUT>
          <TITLE>My-Scan-2</TITLE>
          <USER>
            <USERNAME>qualys_joe</USERNAME>
          </USER>
          <SCHEDULED>0</SCHEDULED>
          <SCAN_DATETIME>2020-08-06 03:52:30</SCAN_DATETIME>
          <NETWORK>
            <ID>1000</ID>
            <NAME>My-Custom-Network</NAME>
          </NETWORK>
          <OPTION_PROFILE>
            <ID>2134</ID>
            <NAME>Initial Options</NAME>
          </OPTION_PROFILE>
          <TARGETS>
          </TARGETS>
        </SCAN_INPUT>
        <SCAN_DETAILS>
          <STATUS>CANCELED</STATUS>
          <LAUNCH_DATETIME>2020-08-06 03:52:30</LAUNCH_DATETIME>
          <DURATION>10</DURATION>
        </SCAN_DETAILS>
      </SCAN_SUMMARY>
    </SCAN_SUMMARY_LIST>
  </RESPONSE>
</SCAN_SUMMARY_OUTPUT>
```
<SCAN_DETAILS>
<SCAN_RESULTS>
<HOSTS>
<COUNT>4</COUNT>
<HOSTS_DATA>
<CANCELLED>
<IP_LIST>
<COUNT>2</COUNT>
<IP_DATA>
<IP_CSV>10.10.25.232, 10.10.25.240</IP_CSV>
</IP_DATA>
</IP_LIST>
<DNS_LIST>
<COUNT>2</COUNT>
<DNS_DATA>
<DNS_CSV>dns1.qualys.com,dns2.qualys.com</DNS_CSV>
</DNS_DATA>
</DNS_LIST>
</CANCELLED>
</HOSTS_DATA>
</HOSTS>
</DETECTIONS>
<IG>
<TOTAL_COUNT>0</TOTAL_COUNT>
<COUNT_BY_SEVERITY>
<SEVERITY_1>0</SEVERITY_1>
<SEVERITY_2>0</SEVERITY_2>
<SEVERITY_3>0</SEVERITY_3>
<SEVERITY_4>0</SEVERITY_4>
<SEVERITY_5>0</SEVERITY_5>
</COUNT_BY_SEVERITY>
</IG>
<VULN>
<CONFIRMED>
<TOTAL_COUNT>0</TOTAL_COUNT>
<COUNT_BY_SEVERITY>
<SEVERITY_1>0</SEVERITY_1>
<SEVERITY_2>0</SEVERITY_2>
<SEVERITY_3>0</SEVERITY_3>
<SEVERITY_4>0</SEVERITY_4>
<SEVERITY_5>0</SEVERITY_5>
</COUNT_BY_SEVERITY>
</CONFIRMED>
<POTENTIAL>
<TOTAL_COUNT>0</TOTAL_COUNT>
<COUNT_BY_SEVERITY>
<SEVERITY_1>0</SEVERITY_1>
<SEVERITY_2>0</SEVERITY_2>
<SEVERITY_3>0</SEVERITY_3>
<SEVERITY_4>0</SEVERITY_4>
<SEVERITY_5>0</SEVERITY_5>
</COUNT_BY_SEVERITY>
</POTENTIAL>
</VULN>

105
Scans
VM Scan Summary

</DETECTIONS>
</SCAN_RESULTS>
</SCAN_SUMMARY>
</SCAN_SUMMARY_LIST>
</RESPONSE>
</SCAN_SUMMARY_OUTPUT>

DTD
<platform API server>/api/2.0/fo/scan/vm/summary/output.dtd
Scan Summary

/ api/2.0/fo/scan/summary/

[GET] [POST]

This is the original VM Scan Summary API for identifying hosts that were not scanned and why. We recommend you try the new improved VM Scan Summary API which has more filter options and enhanced output content. See VM Scan Summary.

Permissions

Manager role is required.

How it works

First we’ll find all the scans launched since the date (or within the date range) that you specify. Then we’ll identify hosts that were included in the scan target but not scanned for some reason. For each host you’ll see the category/reason it was not scanned and the host’s tracking method.

Categories for hosts not scanned:

Excluded - The hosts were excluded. Hosts may be excluded on a per scan basis (by the user launching or scheduling the scan) or globally for all scans. Managers and Unit Managers have privileges to edit the global excluded hosts list for the subscription.

Cancelled - Hosts were not scanned because the scan was cancelled. Scans may be cancelled by a user, by an administrator or automatically by the service as specified in scheduled scan settings.

Dead - The hosts were not “alive” at the time of the scan, meaning that they did not respond to probes sent by the scanning engine, and the option to Scan Dead Hosts was not enabled.

Unresolved - Hosts were scanned but they could not be reported because the NetBIOS or DNS hostname, whichever tracking method is specified for each host, could not be resolved.

Duplicate - The hosts were duplicated within a single segment/slice of the scan job. For example, two different hostnames resolving to the same IP with tracking by IP.

Not Vulnerable - Hosts were found to be not vulnerable during host discovery without having to run a full scan. This could happen for example if the list of QIDs to be scanned are limited to certain ports and those ports are found to be closed.

Aborted - The scan was abruptly discontinued. This is a rare occurrence that may be caused for different reasons. For example, it’s possible that a connection timed out or there were connection errors on a particular port or the scan time elapsed.

Blocked - Hosts were blocked from scanning for some reason. For example, user provided blacklisted IPs to scan and after the scan was launched it was blocked due to improper configuration.
Sample - VM scan summary

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo(scan/summary/?action=list
&scan_date_since=2018-04-27&include_excluded=1&include_unresolved=1
&include_cancelled=1&include_notvuln=1&include_duplicate=1"
```

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCAN_SUMMARY_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/summary/scan_summary
```

---

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>action</strong>=list</td>
<td>(Required)</td>
</tr>
<tr>
<td><strong>scan_date_since</strong>={value}</td>
<td>(Required) Include scans started since a certain date. Specify the date in YYYY-MM-DD format. The date must be less than or equal to today’s date.</td>
</tr>
<tr>
<td><strong>scan_date_to</strong>={value}</td>
<td>(Optional) Include scans started up to a certain date. Specify the date in YYYY-MM-DD format. The date must be more than or equal to scan_date_since, and less than or equal to today’s date.</td>
</tr>
<tr>
<td><strong>output_format</strong>={value}</td>
<td>(Optional) The output format: XML (the default), CSV or JSON.</td>
</tr>
<tr>
<td><strong>tracking_method</strong>={value}</td>
<td>(Optional) By default hosts with any tracking method will be returned in the output. Use this option to only include hosts with a certain tracking method. Valid values are: IP, DNS, NETBIOS.</td>
</tr>
<tr>
<td><strong>include_dead</strong>={0</td>
<td>1}</td>
</tr>
<tr>
<td><strong>include_excluded</strong>={0</td>
<td>1}</td>
</tr>
<tr>
<td><strong>include_unresolved</strong>={0</td>
<td>1}</td>
</tr>
<tr>
<td><strong>include_cancelled</strong>={0</td>
<td>1}</td>
</tr>
<tr>
<td><strong>include_notvuln</strong>={0</td>
<td>1}</td>
</tr>
<tr>
<td><strong>include_blocked</strong>={0</td>
<td>1}</td>
</tr>
<tr>
<td><strong>include_duplicate</strong>={0</td>
<td>1}</td>
</tr>
<tr>
<td><strong>include_aborted</strong>={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
<SCAN_SUMMARY_OUTPUT>
<RESPONSE>
  <DATETIME>2018-05-02T10:45:40Z</DATETIME>
  <SCAN_SUMMARY_LIST>
    <SCAN_SUMMARY>
      <SCAN_REF>scan/1525251885.92469</SCAN_REF>
      <SCAN_DATE>2018-05-02T09:04:34Z</SCAN_DATE>
      <HOST_SUMMARY category="notvuln" tracking="IP">10.10.10.10-10.10.10.15,10.10.10.17</HOST_SUMMARY>
      <HOST_SUMMARY category="notvuln" tracking="DNS">gfi-31-1.caac125.qualys.com,gfi-31-2.caac125.qualys.com</HOST_SUMMARY>
      <HOST_SUMMARY category="notvuln" tracking="NETBIOS">gfi-31-3,gfi-31-4</HOST_SUMMARY>

      <HOST_SUMMARY category="cancelled" tracking="IP">10.10.10.20,10.10.10.22</HOST_SUMMARY>
      <HOST_SUMMARY category="cancelled" tracking="DNS">gfi-31-5.caac125.qualys.com,gfi-31-6.caac125.qualys.com</HOST_SUMMARY>
      <HOST_SUMMARY category="dead" tracking="IP">10.10.10.25</HOST_SUMMARY>
      <HOST_SUMMARY category="dead" tracking="NETBIOS">gfi-31-10,gfi-31-11</HOST_SUMMARY>
      <HOST_SUMMARY category="excluded" tracking="IP">10.10.10.26</HOST_SUMMARY>
      <HOST_SUMMARY category="unresolved" tracking="NETBIOS">gfi-31-13</HOST_SUMMARY>
      <HOST_SUMMARY category="duplicate" tracking="IP">10.10.10.27</HOST_SUMMARY>
      <HOST_SUMMARY category="duplicate" tracking="DNS">gfi-31-14.caac125.qualys.com</HOST_SUMMARY>
    </SCAN_SUMMARY>
  </SCAN_SUMMARY_LIST>
</RESPONSE>
</SCAN_SUMMARY_OUTPUT>

DTD
<platform API server>/api/2.0/fo/scan/summary/scan_summary_output.dtd
Scanner Details

/api/2.0/fo/scan/scanner

[GET] [POST]

Identify the scanner used to scan a particular IP address at a given time.

Permissions - Manager role is required.

This is supported for vulnerability scans only. This API is especially useful when you’re scanning a large number of IPs using a pool of scanners and you’re not sure which scanner was used to scan a particular host.

The XML output will show the IP address scanned with the scan reference number, scan date, the scanner identifier (external scanner or scanner appliance name), scanner type (extranet or appliance) and scanner software versions.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>scan_date_since</td>
<td>(Required) Include scans started since a certain date. Specify the date in YYYY-MM-DD format. The date must be less than or equal to today’s date.</td>
</tr>
<tr>
<td>scan_date_to</td>
<td>(Optional) Include scans started up to a certain date. Specify the date in YYYY-MM-DD format. The date must be later than or equal to scan_date_since, and less than or equal to today’s date.</td>
</tr>
<tr>
<td>ips</td>
<td>(Required) The IP addresses you want scanner details for. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>output_format=XML</td>
<td>(Optional) The output format: XML (the default).</td>
</tr>
</tbody>
</table>

Sample - List scanner details for certain IPs

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list&ips=10.10.10.2-10.10.10.7,10.10.10.10&scan_date_since=2018-05-24&scan_date_to=2018-09-28" "https://qualysapi.qualys.com/api/2.0/fo/scan/scanner/"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE IP_SCANNERS_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/scan/scanner/scanner_list_output.dtd">```
<IP_SCANNERS_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-11-08T21:49:51Z</DATETIME>
    <IP_SCANNERS_OUTPUT>
      <IP_SCANNED>
        <IP>10.10.10.7</IP>
        <SCAN_REF>scan/1527197914.13102</SCAN_REF>
        <SCANNER_IDENTIFIER>external scanner</SCANNER_IDENTIFIER>
        <SCANNER_TYPE>extranet</SCANNER_TYPE>
        <ML_VERSION>ML-9.7.20-1</ML_VERSION>
        <VULNSIGS_VERSION>VULNSIGS-2.4.182-2</VULNSIGS_VERSION>
      </IP_SCANNED>
      <IP_SCANNED>
        <IP>10.10.10.7</IP>
        <SCAN_REF>scan/1538093810.64913</SCAN_REF>
        <SCAN_DATE>2018-09-28T00:19:25Z</SCAN_DATE>
        <SCANNER_IDENTIFIER>Esxi_4_Network</SCANNER_IDENTIFIER>
        <SCANNER_TYPE>appliance</SCANNER_TYPE>
        <ML_VERSION>ML-9.10.21-1</ML_VERSION>
        <VULNSIGS_VERSION>VULNSIGS-2.4.284-2</VULNSIGS_VERSION>
      </IP_SCANNED>
      <IP_SCANNED>
        <IP>10.10.10.10</IP>
        <SCAN_REF>scan/1538093810.64913</SCAN_REF>
        <SCAN_DATE>2018-09-28T00:19:25Z</SCAN_DATE>
        <SCANNER_IDENTIFIER>Esxi_4_Network</SCANNER_IDENTIFIER>
        <SCANNER_TYPE>appliance</SCANNER_TYPE>
        <ML_VERSION>ML-9.10.21-1</ML_VERSION>
        <VULNSIGS_VERSION>VULNSIGS-2.4.284-2</VULNSIGS_VERSION>
      </IP_SCANNED>
    </IP_SCANNERS_OUTPUT>
  </RESPONSE>
</IP_SCANNERS_LIST_OUTPUT>

DTD

<platform API server>/api/2.0/fo/scan/scanner/scanner_list_output.dtd
Share PCI Scan

The Share PCI Scan API (/api/2.0/fo/scan/pci/) provides an automated way to share (export) finished PCI scans to PCI Merchant accounts and check the export status. A PCI scan is a vulnerability scan that was run with the option profile “Payment Card Industry (PCI) Options”.

Express Lite: This API is available to Express Lite users.

In advance of sharing a PCI scan using the share PCI scan API, the target PCI Merchant account must be already defined as a PCI account link within the API user’s Qualys account. Account links can be defined using the Qualys user interface only.

Permissions - Any user with scan permissions (Manager, Unit Manager or Scanner) can share a PCI scan with one of their own PCI Merchant accounts and obtain share status. The user’s Qualys account must allow access to the PCI scan and must have a link to the target PCI Merchant account.

Share Restriction - The following share restriction applies to all users. One PCI scan can be shared (exported) to one PCI Merchant subscription one time only, assuming the share request is successful. (Note: If a particular scan has been exported to any PCI account in the same PCI Merchant subscription as your PCI account, the scan can’t be exported.) If a share request fails for some reason, it’s possible to submit another share request for the same PCI scan and PCI Merchant account.

Share a PCI Scan

/api/2.0/fo/scan/pci/ with action=share

[POST]

Export a finished PCI scan to a selected PCI Merchant account. It’s possible to export a PCI scan one time per PCI Merchant account, and the same PCI scan can be exported to multiple PCI Merchant accounts.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=share</td>
<td>(Required) Specify &quot;share&quot; to share a PCI scan.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_ref={value}</td>
<td>(Required) The scan reference of a finished PCI scan. The scan status of this scan must be &quot;Finished&quot;.</td>
</tr>
<tr>
<td>merchant_username={value}</td>
<td>(Required) The user name of the PCI Merchant account that the PCI scan will be exported to. The API user’s Qualys account must have a PCI account link already defined for this target PCI Merchant account.</td>
</tr>
</tbody>
</table>
Sample - Share PCI scan

API request:

curl -s -H "X-Requested-With: curl demo 2" -D headers.15 -b "QualysSession=38255848108d68a2feaf9ee664ca78a7; path=/api; secure" -d "action=share&merchant_username=manager1@qualys&scan_ref=scan/1281646610.5720" "https://qualysapi.qualys.com/api/2.0/fo/scan/pci/"

XML output Successful Share:
The XML output uses the simple return DTD and the message is “Requested share of scan to PCI”.

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-01-17T00:50:39Z</DATETIME>
    <TEXT>Requested share of scan to PCI</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>scan_ref</KEY>
        <VALUE>scan/1281646610.5720</VALUE>
      </ITEM>
      <ITEM>
        <KEY>merchant_username</KEY>
        <VALUE>manager1@qualys</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>

XML output Share Already in Progress or Completed:
When the request to share a PCI scan fails, the XML output uses the simple return DTD with the error. If the failure is because sharing is in progress for the PCI Merchant account or the scan has already been shared to the PCI account, the output includes the message “This scan has already been shared with the Merchant account”.

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-01-04T14:54:01Z</DATETIME>
    <CODE>999</CODE>
  </RESPONSE>
</SIMPLE_RETURN>
<TEXT>This scan has already been shared with the Merchant account.</TEXT>
</RESPONSE>
</SIMPLE_RETURN>

Get PCI Share Status

/api/2.0/fo/scan/pci/ with action=status

[GET] [POST]

Get the share status of a PCI scan that has already been shared with a PCI merchant account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=status</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_ref={value}</td>
<td>(Required) The scan reference of the shared scan that you want to check the export status for.</td>
</tr>
<tr>
<td>merchant_username= {value}</td>
<td>(Required) The username of the PCI account which the scan was shared with.</td>
</tr>
</tbody>
</table>

Sample - PCI Share status

API request:

curl -s -H "X-Requested-With: curl demo 2" -u "USERNAME:PASSWD" -d "action=status&scan_ref=scan/1531755831.21639&merchant_username=asmith@hq" "https://qualysapi.qualys.com/api/2.0/fo/scan/pci/

XML output:
The XML response for a status requests identifies the share status: Queued (request was received and not started yet), In Progress, Finished (scan was exported to PCI account successfully), or Error.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE PCI_SCAN_SHARE_STATUS SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/scan/pci/pici_scan_share_status.dtd">
<PCI_SCAN_SHARE_STATUS>
  <RESPONSE>
    <SCAN>
      <MERCHANT_USERNAME>asmith@hq</MERCHANT_USERNAME>
      <SCAN_REF>scan/1531755831.21639</SCAN_REF>
    </SCAN>
  </RESPONSE>
</PCI_SCAN_SHARE_STATUS>
```
<STATUS>In Progress</STATUS>
<LAST_SHARED>2018-07-19T05:55:58Z</LAST_SHARED>
</SCAN>
</RESPONSE>
</PCI_SCAN_SHARE_STATUS>

**DTD**

<platform API server>/api/2.0/fo/scan/pci/pci_scan_share_status.dtd
Discovery Scans (maps)

Launch discovery scans, also called maps, to launch network discovery of your domains and/or IP addresses in asset groups. This returns an inventory of your network devices.

Launch Map | Map Report List | Cancel Running Map | Download Saved Map Report | Delete Saved Map Report | Domain List | Add/Edit Domain

Launch Map

/msp/map-2.php

[GET] [POST]

Launch a Qualys network map for one or more domains, initiating network discovery. The map target may include asset groups and the default scanner option may be enabled for distributed mapping across multiple scanner appliances.

Basic HTTP authentication is required. Session based authentication is not supported using this API.

A map request for multiple domains issued using the map-2.php API, runs one map at a time, one domain at a time. If you cancel a running map for a domain using the scan_cancel.php function and there are multiple domains in the map target, the service cancels the maps for any remaining, undiscovered domains in the same map target.

For a map request with multiple domains, the XML map report returned by the map-2.php function includes all domains that were successfully discovered. When you view the map results for this request using the map_report.php function or the Qualys user interface, each map report includes map results for one domain. Also, if the map summary notification is enabled in your account, there is a separate notification for each target domain.

Permissions - Managers can map all domains in the subscription. Unit Managers can map domains in the user’s same business unit. Scanners can map domains in their own account.
## Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>map_title={title}</td>
<td>(Optional) Specifies a title for the map. The map title can have a maximum of 2,000 characters. When specified, the map title appears in the header section of the map results. When unspecified, the API returns a standard, descriptive title in the header section.</td>
</tr>
<tr>
<td>domain={target}</td>
<td>(Optional) Specifies one or more domain names for the map target. Multiple entries are comma separated. (Target may include domain names and/or asset groups) For each domain, include the domain name only; do not enter “www.” at the start of the domain name. Netblocks may be specified with each domain name to extend the scope of the map. Multiple domains must be comma separated. This parameter and/or asset_groups must be specified.</td>
</tr>
<tr>
<td>asset_groups={title1,title2...}</td>
<td>(Optional) Specifies the titles of asset groups for the map target. Multiple asset groups must be comma separated. (Target may include domain names and/or asset groups) This parameter and/or the domain parameter must be specified.</td>
</tr>
<tr>
<td>isscanner_name={name}</td>
<td>(Optional) Specifies the name of the Scanner Appliance for the map, when the map target has private use internal IPs. Using Express Lite, Internal Scanning must be enabled in your account. One of these parameters may be specified in the map request: isscanner_name or default scanner.</td>
</tr>
<tr>
<td>default_scanner=1</td>
<td>(Optional) Enables the default scanner feature, which is only valid when the map target consists of asset groups. A valid value is 1 to enable the default scanner, or 0 (the default) to disable it. Using Express Lite, Internal Scanning must be enabled in your account. One of these parameters may be specified in the same map request: isscanner_name or default scanner.</td>
</tr>
<tr>
<td>option={title}</td>
<td>(Optional) Specifies the title of an option profile to be applied to the map. The profile title must be defined in the user account, and it can have a maximum of 64 characters. If unspecified, the default option profile in the user account is applied.</td>
</tr>
</tbody>
</table>
Scans
Discovery Scans (maps)

Request a map of the domain "www.mycompany.com" using the external scanners and to receive a map report:


Request a map of the domain "www.mycompany.com" using the external scanners, save map report on the Qualys platform:


Request a map for the following domain/netblock pair using the scanner appliance “Hong Kong” and custom domain mycompany:


Request a map for this domain/netblock pair using the scanner appliance “San Francisco” and none domain:


Request a map for the domain “mydomain.com” using a network ID and to receive a map report:


<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network_id={value}</td>
<td>(Optional, and valid only when the Network Support feature is enabled for the user’s account) Restrict the request to a certain custom network by specifying the network ID. When unspecified, we default to “0” for Global Default Network.</td>
</tr>
<tr>
<td>save_report=yes</td>
<td>(Optional) Saves a map report for each target domain on the Qualys server for later use. A valid value is “yes” to save a map report for each target domain, or “no” (the default) to not save the report. If set to “yes”, you can close the HTTP connection when the map is in progress, without cancelling the map. When the map completes the resulting map report is saved on the Qualys platform, and a map summary email notification is sent (if this option is enabled in your user account). Saved map reports can be retrieved using map_report_list.php and map_report.php.</td>
</tr>
</tbody>
</table>
Map Report List

/msp/map_report_list.php
[GET] [POST]

List saved map reports in the user’s account. Each entry in the map report list identifies a saved map report for a specific domain. There is a separate saved map report for each domain in the map target.

Basic HTTP authentication is required. Session based authentication is not supported using this API.

Permissions - Managers can view all saved map reports in the subscription. Unit Managers can view saved map reports for domains in user’s business unit. Scanners and Readers can view saved map reports for domains in user’s account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>last=yes</td>
<td>(Optional) Used to retrieve information only about the last saved map report. A valid value is &quot;yes&quot; to retrieve the last saved map report, or &quot;no&quot; (the default) to retrieve all map reports.</td>
</tr>
<tr>
<td>domain={target}</td>
<td>(Optional) Used to receive a list of all saved map reports for the specified target domain. If both parameters domain={target} and last=yes are specified, you will receive information about the last saved map for the target domain.</td>
</tr>
</tbody>
</table>

Sample

Receive information about the last saved map for the domain "www.companyabc.com":

domain=www.companyabc.com&last=yes

DTD

<platform API server>/map-report_list.dtd
**Running Map Report List**

/msp/scan_running_list.php

[GET]  [POST]

List maps and scans that are currently running in the user’s account. If you’re interested in listing scans only (not maps), we recommend using VM Scan List (/api/2.0/fo/scan/) instead.

Basic HTTP authentication is required. Session based authentication is not supported using this API.

Permissions - Managers can view all running maps/scans in the subscription. Unit Managers can view running maps/scans on assets in the user’s business unit. Scanners and Readers can view running maps/scans on assets their account.

**Sample - Running map/scan list**

https://qualysapi.qualys.com/msp/scan_running_list.php?

**DTD**

<platform API server>/scan_running_list.dtd

**Cancel Running Map**

/msp/scan_cancel.php

[GET]  [POST]

Cancel a map in progress. It’s not possible to cancel a map when it has the scan status “Loading”.

Basic HTTP authentication is required. Session based authentication is not supported using this API.

Permissions - Managers can cancel all running maps in the subscription. Unit Managers can cancel running maps launched by users in their same business unit. Scanners can cancel running maps they have launched.

**Input Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref={value}</td>
<td>(Required) Specifies the map reference for the map to be cancelled (or a scan reference for the scan to be cancelled). A map reference starts with “map/”.</td>
</tr>
</tbody>
</table>
Sample - Cancel a map in progress

DTD
<platform API server>/generic_return.dtd

Download Saved Map Report
/msp/map_report.php
[GET] [POST]

Download a saved map in the user’s account, when the map has the scan status “Finished”. Each saved map report identifies map results for a specific domain. If you issue a map request for multiple domains using the map-2.php API, there is a separate saved map report for each domain in the map target.

Basic HTTP authentication is required. Session based authentication is not supported using this API.

Permissions - Managers can download all saved map reports in subscription. Unit Managers can download saved map report for domain in user’s business unit. Scanners and Readers can download saved map report for domain in user’s account.

Input Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref={value}</td>
<td>(Required) Specifies the map reference for the scan you want to download. A map reference starts with “map/”.</td>
</tr>
</tbody>
</table>

Sample - Download saved map report
https://qualysapi.qualys.com/msp/map_report.php?
ref=map/987659876.19876

DTD
<platform API server>/map.dtd
Delete Saved Map Report

/msp/scan_report_delete.php

[GET]  [POST]

Delete a previously saved network map or scan report, when the scan status is “Finished”.

Basic HTTP authentication is required. Session based authentication is not supported using this API.

Permissions - Managers can delete saved map reports in the subscription. Unit Managers can delete saved map reports for domains in the user’s business unit, including the user’s own maps and maps run by other users in the same business unit. Scanners can delete saved map reports in user’s account.

Input Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref={value}</td>
<td>(Required) Specifies the map reference for the map to be deleted. A map reference starts with &quot;map/&quot;.</td>
</tr>
</tbody>
</table>

Sample - Delete saved map report

ref=map/999666888.12345

DTD

<platform API server>/generic_return.dtd
**Domain List**

/msp/asset_domain_list.php

[GET]  [POST]

List asset domains in the user account.

Basic HTTP authentication is required. Session based authentication is not supported using this API.

Permissions - Managers can view all domains in subscription. Unit Managers can view domains in user's business unit. Scanners, Readers can view domains in their own account.

**Sample - List all domains in account**

  https://qualysapi.qualys.com/msp/asset_domain_list.php

**DTD**

<platform API server>/domain_list.dtd

**Add/Edit Domain**

/msp/asset_domain.php

[GET]  [POST]

Add and edit domains and related netblocks in the subscription. The domains defined may be used as targets for network scans (maps).

Basic HTTP authentication is required. Session based authentication is not supported using this API.

Permissions - Manager user role is required.

**Input Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={add</td>
<td>edit}</td>
</tr>
</tbody>
</table>
### Sample - Add domain


### Sample - Edit domain

https://qualysapi.qualys.com/msp/asset_domain.php?action=edit&domain=acme.com&netblock=10.10.0.1/24,10.1.1.0-10.1.1.100

### DTD

<platform API server>/generic_return.dtd
Scan Configuration

Manage scan configurations in your account - scanner appliances, KnowledgeBase, search lists and option profiles.

Scanner Appliance List
Manage Virtual Scanner Appliances
Update Physical Scanner Appliance
Replace Scanner Appliance
Scanner Appliance VLANs and Static Routes
Option Profile Export | Option Profile Import
Option Profiles for VM | PCI | PC
KnowledgeBase | Editing Vulnerabilities
Static Search Lists
Dynamic Search Lists | Vendor IDs and References
Scanner Appliance List

/api/2.0/fo/appliance/?action=list

[GET] [POST]

List scanner appliances in your account with their configurations. The list output is shown in “brief” mode by default. Specify output_mode=full to include full output (the same information available within the Qualys user interface).

Permissions - Managers can view all scanner appliances in the subscription. Unit Managers can view appliances in the user’s own business unit. Scanners and Readers can view appliances in their own account.

Express Lite - This API is available to Express Lite users when Internal Scanning is enabled in the user’s account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required) A flag used to make a request for a list of scanner appliances. The GET or POST method may be used for a list request.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>output_mode={brief</td>
<td>full}</td>
</tr>
<tr>
<td>scan_detail={0</td>
<td>1}</td>
</tr>
<tr>
<td>show_tags={0</td>
<td>1}</td>
</tr>
<tr>
<td>include_cloud_info={0</td>
<td>1}</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>busy=[0</td>
<td>1]</td>
</tr>
<tr>
<td>scan_ref=[value]</td>
<td>(Optional) Specify a scan reference code to show only the scanner appliances running a particular scan. You may enter a valid scan reference code for a currently running scan.</td>
</tr>
<tr>
<td>name=[string]</td>
<td>(Optional) List only scanner appliances (physical and virtual) that have names matching the string provided. Tip - Substring match is supported. For example, if you have 2 appliances named “myscanner” and “anotherscanner” and you supply the string “name=scan” both appliance both appliances will be returned in the XML output.</td>
</tr>
<tr>
<td>ids=[id1,id2,...]</td>
<td>(Optional) List only scanner appliances (physical and virtual) that have certain IDs. Multiple IDs are comma separated.</td>
</tr>
<tr>
<td>include_license_info=[0</td>
<td>1]</td>
</tr>
<tr>
<td>type={physical</td>
<td>virtual</td>
</tr>
<tr>
<td>platform_provider</td>
<td>(Optional) Specify a platform to show scanners deployed on that platform. The valid values are: ec2, ec2_compat, gce, azure, vCenter.</td>
</tr>
<tr>
<td></td>
<td>ec2 - Amazon EC2, ec2_compat - OpenStack, gce - Google Cloud Platform, azure - Microsoft Azure Cloud Platform, vCenter - VMware vCenter</td>
</tr>
</tbody>
</table>

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list&echo_request=1&ids=777,1127,1131&include_license_info=1" "https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE APPLIANCE_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/appliance/appliance_list_output.dtd">
<APPLIANCE_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2014-01-02T09:26:01Z</DATETIME>
    <APPLIANCE_LIST>
      <APPLIANCE>
        <ID>777</ID>
        <NAME>scanner1</NAME>
        <SOFTWARE_VERSION>2.6</SOFTWARE_VERSION>
        <RUNNING_SCAN_COUNT>0</RUNNING_SCAN_COUNT>
        <STATUS>Online</STATUS>
      </APPLIANCE>
      <APPLIANCE>
        <ID>1127</ID>
        <NAME>scanner2</NAME>
        <SOFTWARE_VERSION>2.6</SOFTWARE_VERSION>
        <RUNNING_SCAN_COUNT>0</RUNNING_SCAN_COUNT>
        <STATUS>Online</STATUS>
      </APPLIANCE>
      <APPLIANCE>
        <ID>1131</ID>
        <NAME>scanner3</NAME>
        <SOFTWARE_VERSION>2.6</SOFTWARE_VERSION>
        <RUNNING_SCAN_COUNT>0</RUNNING_SCAN_COUNT>
        <STATUS>Offline</STATUS>
      </APPLIANCE>
    </APPLIANCE_LIST>
    <LICENSE_INFO>
      <QVSA_LICENSES_COUNT>10</QVSA_LICENSES_COUNT>
      <QVSA_LICENSES_USED>3</QVSA_LICENSES_USED>
    </LICENSE_INFO>
  </RESPONSE>
</APPLIANCE_LIST_OUTPUT>
```
API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list&type=virtual&platform_provider=ec2&include_cloud_info=1&output_mode=full" "https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

XML output:

Sample shows Cloud Info for Amazon EC2.

```xml
```
API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list&output_mode=full" "https://qualysapi.qualys.com/api/2.0/fo/appliance/"

XML output:
Sample shows the type of scanner appliance and the specifications regarding the CPU, memory, and region:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE APPLIANCE_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/appliance/appliance_list_output.dtd">
<APPLIANCE_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-08-31T09:14:49Z</DATETIME>
    <APPLIANCE_LIST>
      <APPLIANCE>
        <ID>132455</ID>
        <UUID>6ae4efce-0c5e-e227-82e0-1b7f55f1b98b</UUID>
        <NAME>VS_ND_1</NAME>
        <SOFTWARE_VERSION>2.6</SOFTWARE_VERSION>
        <RUNNING_SLICES_COUNT>0</RUNNING_SLICES_COUNT>
        <RUNNING_SCAN_COUNT>0</RUNNING_SCAN_COUNT>
        <STATUS>Offline</STATUS>
        <MODEL_NUMBER>cvscanner</MODEL_NUMBER>
        <TYPE>Virtual</TYPE>
        <SERIAL_NUMBER>0</SERIAL_NUMBER>
        <ACTIVATION_CODE>15440265032293</ACTIVATION_CODE>
        <INTERFACE_SETTINGS>
          <INTERFACE>lan</INTERFACE>
          <IP_ADDRESS>1.1.1.1</IP_ADDRESS>
          <NETMASK>128.0.0.0</NETMASK>
          <GATEWAY>128.0.0.0</GATEWAY>
          <LEASE>Static</LEASE>
          <IPV6_ADDRESS></IPV6_ADDRESS>
          <SPEED></SPEED>
          <DUPLEX>Unknown</DUPLEX>
          <DNS>
            <DOMAIN></DOMAIN>
            <PRIMARY>128.0.0.0</PRIMARY>
            <SECONDARY>128.0.0.0</SECONDARY>
          </DNS>
        </INTERFACE_SETTINGS>
        <MAX_CAPACITY_UNITS></MAX_CAPACITY_UNITS>
      </APPLIANCE>
      ...
    </APPLIANCE_LIST>
  </RESPONSE>
</APPLIANCE_LIST_OUTPUT>
```
<CPU_INFO>Intel(R) Xeon(R) CPU E5-2699 v3
@2.30GHz</CPU_INFO>

<MEMORY_INFO>2,070,484 MB</MEMORY_INFO>

<REGION_INFO></REGION_INFO>

... 

DTD:
<platform API server>/api/2.0/fo/appliance/appliance_list_output.dtd

Manage Virtual Scanner Appliances

Use the Scanner Appliance API (/api/2.0/fo/appliance/) to create, update and delete virtual scanner appliances.

Tell me about permissions. Managers can perform all actions (create, update, delete). Unit Managers and Scanners must have the “Manage virtual scanner appliances” permission to create, update and delete virtual scanners. This permission is only available to Scanner users when your subscription is configured to allow it.

Add New Virtual Scanner Appliance

/api/2.0/fo/appliance/ with action=create

[POST]

Create a new virtual scanner appliance in your account.

Permissions - Managers can create new virtual scanner appliance. Unit Managers and Scanners must have the “Manage virtual scanner appliances” permission. This permission is only available to Scanner users when your subscription is configured to allow it.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=create</td>
<td>(Required)</td>
</tr>
<tr>
<td>name=[string]</td>
<td>(Required) The friendly name. This name can’t already be assigned to an appliance in your account. It can be a maximum of 15 characters, spaces are not allowed.</td>
</tr>
</tbody>
</table>
## Manage Virtual Scanner Appliances

### API request:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=create&echo_request=1&name=scanner1"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>polling_interval={value}</td>
<td>(Optional) The polling interval, in seconds. A valid value is 60 to 3600 (we recommend 180 which is the default). This is the frequency that the virtual scanner will attempt to connect to our Cloud Security Platform. The appliance calls home to provide health updates/heartbeats to the platform, to get software updates from the platform, to learn if new scan jobs have been requested by users, and to upload scan results data to the platform, if applicable.</td>
</tr>
<tr>
<td>asset_group_id={value}</td>
<td>(Required for Unit Managers and Scanners for Create request) The ID of an asset group the virtual scanner will be assigned to.</td>
</tr>
</tbody>
</table>
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE APPLIANCE_LIST_OUTPUT SYSTEM
  "https://qualysapi.qualys.com/api/2.0/fo/appliance/appliance_create_output.dtd">
<APPLIANCE_CREATE_OUTPUT>
  <RESPONSE>
    <DATETIME>2014-01-02T09:26:01Z</DATETIME>
    <ID>777</ID>
    <NAME>scanner1</NAME>
    <ACTIVATION CODE>ACTIVATION-CODE</ACTIVATION CODE>
    <REMAINING_QVSA_LICENSES>4</REMAINING_QVSA_LICENSES>
  </RESPONSE>
</APPLIANCE_CREATE_OUTPUT>
```

DTD:

```xml
<platform API server>/api/2.0/fo/appliance/appliance_create_output.dtd
```

Update Virtual Scanner Appliance

/`api/2.0/fo/appliance/` with action=update

[POST]

Update a virtual scanner appliance in your account. You can add tags, remove and reset tags for your scanner appliances.

Permissions - Managers can update a virtual scanner appliance. Unit Managers and Scanners must have the “Manage virtual scanner appliances” permission. This permission is only available to Scanner users when your subscription is configured to allow it.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={id}</td>
<td>(Required) A valid ID of a virtual scanner.</td>
</tr>
<tr>
<td>name={string}</td>
<td>(Optional) The friendly name. This name can’t already be assigned to an appliance in your account. It can be a maximum of 15 characters, spaces are not allowed.</td>
</tr>
</tbody>
</table>
**Scan Configuration**  
**Manage Virtual Scanner Appliances**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>polling_interval=</td>
<td>(Optional) The polling interval, in seconds. A valid value is 60 to 3600 (we recommend 180 which is the default). This is the frequency that the virtual scanner will attempt to connect to our Cloud Security Platform. The appliance calls home to provide health updates/heartbeats to the platform, to get software updates from the platform, to learn if new scan jobs have been requested by users, and to upload scan results data to the platform, if applicable.</td>
</tr>
<tr>
<td>comment=</td>
<td>(Optional) User-defined comments.</td>
</tr>
<tr>
<td>set_tags=</td>
<td>(Optional) Specify tag to be assigned to the scanner appliance. Both virtual and physical scanners can be tagged. These parameters are mutually exclusive and cannot be specified in the same request: set_tags and add_tags, remove_tags.</td>
</tr>
<tr>
<td>add_tags=</td>
<td>(Optional) Specify tag to be added to the existing list of tags assigned to the scanner. Multiple entries are comma separated. These parameters are mutually exclusive and cannot be specified in the same request: set_tags and add_tags, remove_tags.</td>
</tr>
<tr>
<td>remove_tags=</td>
<td>(Optional) Specify tag to be removed from the existing list of tags assigned to scanner. Multiple tags are comma separated. These parameters are mutually exclusive and cannot be specified in the same request: set_tags and add_tags, remove_tags.</td>
</tr>
<tr>
<td>tag_set_by=id</td>
<td>(Optional) Specify &quot;id&quot; (the default) to select a tag set by providing tag IDs. Specify &quot;name&quot; to select a tag set by providing tag names.</td>
</tr>
<tr>
<td>enable_ipv6=</td>
<td>Enable or disable IPv6 on LAN. Specify 1 to enable IPv6 and 0 to disable it.</td>
</tr>
</tbody>
</table>

**Sample - Update virtual scanner appliance name**

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"  
-d "action=update&echo_request=1&id=12345&name=scanner15"  
"https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM  
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">  
<SIMPLE_RETURN>  
  <RESPONSE>  
    <DATETIME>2014-04-03T12:12:45Z</DATETIME>  
    <TEXT>Virtual scanner updated successfully</TEXT>  
  </RESPONSE>  
</SIMPLE_RETURN>
```
Scan Configuration
Manage Virtual Scanner Appliances

API request:

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
RESPONSE>
<DATETIME>2016-09-15T19:44:35Z</DATETIME>
<TEXT>Virtual scanner updated successfully</TEXT>
ITEM_LIST>
ITEM>
KEY>ID</KEY>
VALUE>3105</VALUE>
ITEM>
ITEM_LIST>
/SIMPLE_RETURN>

Sample - Add tags for windows agent, remove tags for linux agents

API request:

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
RESPONSE>
<DATEIME>2016-09-15T19:44:35Z</DATEIME>
<TEXT>Virtual scanner updated successfully</TEXT>
ITEM_LIST>
ITEM>
KEY>ID</KEY>
VALUE>3112</VALUE>
ITEM>
ITEM_LIST>
/SIMPLE_RETURN>

Sample - Assign tags to virtual scanner appliance

API request:

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
RESPONSE>
<DATEIME>2016-09-15T19:44:35Z</DATEIME>
<TEXT>Virtual scanner updated successfully</TEXT>
ITEM_LIST>
ITEM>
KEY>ID</KEY>
VALUE>3112</VALUE>
ITEM>
ITEM_LIST>
/SIMPLE_RETURN>
Delete Virtual Scanner Appliance

/api/2.0/fo/appliance/ with action=delete
[POST]

Delete a virtual scanner appliance in your account.

Permissions - Managers can delete new virtual scanner appliance. Unit Managers and Scanners must have the “Manage virtual scanner appliances” permission. This permission is only available to Scanner users when your subscription is configured to allow it.

Deleting a virtual scanner results in these actions: 1) The scanner will be removed from associated Asset Groups, and 2) Scheduled Scans using this scanner will be deactivated.

Is your virtual scanner running scans? If yes it's not possible to delete it. We recommend you check to be sure the virtual scanner you want to delete is not running scans.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={id}</td>
<td>(Required) A valid ID of a virtual scanner.</td>
</tr>
</tbody>
</table>

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=delete&echo_request=1&id=12345" "https://qualysapi.qualys.com/api/2.0/fo/appliance/"

XML output:
The XML output uses the simple return (/api/2.0/simple_return.dtd).

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE APPLIANCE_LIST_OUTPUT SYSTEM...>
"https://qualysapi.qualys.com/api/2.0/fo/simple_return.dtd">

  <RESPONSE>
    <DATETIME>2014-01-02T09:26:01Z</DATETIME>
    <TEXT>Virtual scanner deleted successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>115</VALUE>
      </ITEM>
      <ITEM>
        <KEY>DEACTIVATED_SCHEDULED_SCANS</KEY>
        <VALUE>None</VALUE>
      </ITEM>
      <ITEM>
        <KEY>AFFECTED_ASSET_GROUPS</KEY>
        <VALUE>None</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>

Update Physical Scanner Appliance

/api/2.0/fo/appliance/physical/ with action=update

[POST]

Using the Physical Scanner Appliance API (/api/2.0/fo/appliance/physical/), Managers and
Unit Managers can update physical scanner appliances.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update</td>
<td>(Required)</td>
</tr>
<tr>
<td>id=[id]</td>
<td>(Required) A valid ID of a physical scanner.</td>
</tr>
<tr>
<td>name=[string]</td>
<td>(Optional) The friendly name. This name can’t already be assigned to an appliance in your account. It can be a maximum of 15 characters, spaces are not allowed.</td>
</tr>
</tbody>
</table>
Sample 1

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=update&id=5115&comment=Hello"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/physical/"
```

Sample 2

Add VLAN and routes with Name, Polling interval and comments to Physical scanner:

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X POST -d
```
"action=update&id=5115&name=physscanner&polling_interval=360&set_routes=10.10.10.10|255.255.255.0|10.10.10.10|routes1&set_vlans=1|10.2.0.2|255.255.255.0|Testvlan1&comment=Update_scanner"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/physical/"

Sample 3
Update physical scanner using tag_set_by and add_tags parameters:

API Request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=update&id=5115&tag_set_by=id&add_tags=7691422"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/physical/"

Sample 4
Update physical scanner using tag_set_by and set_tags parameters:

API Request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=update&id=5115&tag_set_by=id&set_tags=7691422"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/physical/"

Sample 5
Update physical scanner using tag_set_by and remove_tags parameters:

API Request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=update&id=5115&tag_set_by=id&remove_tags=7691422"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/physical/"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2017-10-01T00:12:29Z</DATETIME>
    <TEXT>Physical scanner updated successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>5115</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
Replace Scanner Appliance

Using the Replace Scanner Appliance API (/api/2.0/fo/appliance/replace_iscanner), Managers and Unit Managers can replace a scanner appliance with a new one. Tell us the name of the appliance you want to replace and the one you want to use.

Good to Know
- You can replace one scanner appliance at a time.
- Do not replace a scanner appliance while scans (using the appliance) are in progress.
- The old scanner and the new scanner must be in the same network, if applicable.
- You can only replace an EC2 scanner with another EC2 scanner.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=replace</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>old_scanner_name={value}</td>
<td>(Required) The name of the scanner you want to replace.</td>
</tr>
<tr>
<td>new_scanner_name={value}</td>
<td>(Required) The name of the scanner you want to use.</td>
</tr>
<tr>
<td>do_not_copy_settings={0</td>
<td>1}</td>
</tr>
<tr>
<td>do_not_remove_new_scanner_from_objects={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

Sample - Replace scanner with new one

Replace “scanner1” with “scanner2” and copy scanner appliance settings but do not remove the new scanner from business objects.

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" "https://qualysapi.qualys.com/api/2.0/fo/appliance/replace_iscanner/?action=replace&echo_request=1&old_scanner_name=scanner1&new_scanner_name=scanner2&do_not_copy_settings=0&do_not_remove_new_scanner_from_objects=1"
```
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCANNER_REPLACE_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/appliance/replace_iscanner/replace_iscanner_output.dtd">
<SCANNER_REPLACE_OUTPUT>
  <REQUEST>
    <DATETIME>2018-01-16T06:52:53Z</DATETIME>
    <USER_LOGIN>abcd</USER_LOGIN>

    <RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/appliance/replace_iscanner/</RESOURCE>
    <PARAM_LIST>
      <PARAM>
        <KEY>echo_request</KEY>
        <VALUE>1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>old_scanner_name</KEY>
        <VALUE>scanner1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>new_scanner_name</KEY>
        <VALUE>scanner2</VALUE>
      </PARAM>
      <PARAM>
        <KEY>do_not_copy_settings</KEY>
        <VALUE>0</VALUE>
      </PARAM>
      <PARAM>
        <KEY>do_not_remove_new_scanner_from_objects</KEY>
        <VALUE>1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>action</KEY>
        <VALUE>replace</VALUE>
      </PARAM>
    </PARAM_LIST>
  </REQUEST>

  <RESPONSE>
    <DATETIME>2018-01-16T06:52:53Z</DATETIME>
    <NEW_SETTINGS>POLLING_INTERVAL: 180, HEARTBEAT: 1</NEW_SETTINGS>
    <SCHEDULED_SCANS>Scheduled-Scan1, Scheduled-Scan2</SCHEDULED_SCANS>
    <ASSET_GROUPS>AG123, AG456</ASSET_GROUPS>
  </RESPONSE>
</SCANNER_REPLACE_OUTPUT>
```
<SUCCESS>Scanner Appliance replaced successfully.</SUCCESS>
</RESPONSE>
</SCANNER_REPLACE_OUTPUT>

**DTD**

A replace scanner appliance API request uses this DTD:

```
<platform API
server>/api/2.0/fo/appliance/replace_iscanner/replace_iscanner_output.dtd
```

**Scanner Appliance VLANs and Static Routes**

/\api/2.0/fo/appliance/?action=update (virtual appliance)

/\api/2.0/fo/appliance/physical/?action=update (physical appliance)

Manage your VLANs and static routes for virtual and physical scanner appliances using the Virtual Scanner Appliance API (/\api/2.0/fo/appliance/physical/?action=update). Use the parameters “set_vlans” and “set_routes” to add, update and remove these settings.

What do I need? Your Qualys account must have the VLANs and Static Routes feature enabled. Please contact our Support Team or your Qualys TAM if you would like us to enable this feature for you.

Permissions - Managers can add/remove VLANs and static routes for all scanner appliances in the subscription. Unit Managers can add/remove VLANs and static routes in the user’s same business unit.

**Set VLANs on Scanner Appliance**

Use the “set_vlans” parameter to specify one or more VLANs.

The format for a single VLAN is ID|IPv4_ADDRESS|NETMASK|NAME|ipv6_static or ipv6_auto|IPv6_ADDRESS, with pipe (|) used as a delimiter. All attributes are required. Multiple VLANs can be assigned using a comma separated list.

Good to know - An API call with the parameter “set_vlans” set to " (empty string) will replace (i.e. remove) *all* of the VLANs that are assigned to the scanner appliance.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Customer-defined ID (not assigned by Qualys). Must be in the range 0 to 4096, inclusive.</td>
</tr>
<tr>
<td>IPv4_ADDRESS</td>
<td>A valid IPv4 IP address (dotted quad), such as 10.10.10.1. Leave empty when specifying an IPv6 address.</td>
</tr>
<tr>
<td>NETMASK</td>
<td>A valid network mask (dotted quad), such as 255.255.255.0. Leave empty when specifying an IPv6 address.</td>
</tr>
</tbody>
</table>
### Attribute Description

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>A valid name (can be empty). The name can be a maximum of 256 ASCII characters. The character : (colon) is permitted. These characters are not permitted: , (comma), &lt; (less than), &gt; (greater than), &quot; (double quote), &amp; (ampersand),</td>
</tr>
<tr>
<td>ipv6_static or ipv6_auto</td>
<td>Specify ipv6_static to provide a static IPv6 address. Specify ipv6_auto to auto-configure IPv6 using SLAAC on the VLAN.</td>
</tr>
<tr>
<td>IPv6_ADDRESS</td>
<td>A valid IPv6 address is required when ipv6_static is specified, such as fdd1:0:1:107::500. Leave empty when ipv6_auto is specified.</td>
</tr>
</tbody>
</table>

**API request (1 IPv4 VLAN):**

```bash
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "id=43463&set_vlans=0|10.10.10.1|255.255.255.0|vlan1"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

**API request (mix of IPv6 and IPv4 VLANs):**

```bash
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "id=43463&set_vlans=1234|||Name1234|ipv6_static|fdd1:0:1:108::500,5678|123.123.123.123|255.255.255.255|Name5678,9012|244.244.244.244|255.255.255.0|Name9012|ipv6_auto,3456|12.12.12.12|255.255.255.0|Name3456|ipv6_static|fdd1:0:1:107::500"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-07-09T08:46:54Z</DATETIME>
    <TEXT>Virtual scanner updated successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>43463</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
**Set Static Routes on Scanner Appliance**

Use the “set_routes” parameter to specify one or more static routes.

The format for a single static route is **IPv4_ADDRESS|NETMASK|IPv4_GATEWAY|NAME|IPv6_ADDRESS|IPv6_GATEWAY**, with pipe (|) used as the delimiter. All attributes are required. Multiple static routes can be assigned using a comma separated list.

Good to know - An API call with the parameter “set_routes” set to " (empty string) will replace (i.e. remove) *all* of the static routes that are assigned to the scanner appliance.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4_ADDRESS</td>
<td>A valid IPv4 IP address (dotted quad), such as 10.10.26.0. Leave empty when specifying an IPv6 address.</td>
</tr>
<tr>
<td>NETMASK</td>
<td>A valid network mask (dotted quad), such as 255.255.255.0. Leave empty when specifying an IPv6 address.</td>
</tr>
<tr>
<td>IPv4_GATEWAY</td>
<td>A valid IPv4 address (dotted quad), such as 10.10.25.255. Leave empty when specifying an IPv6 address.</td>
</tr>
<tr>
<td>NAME</td>
<td>A valid name (can be empty). The name can be a maximum of 256 ASCII characters. The character : (colon) is permitted. These characters are not permitted: , (comma), &lt; (less than), &gt; (greater than), &quot; (double quote), &amp; (ampersand),</td>
</tr>
<tr>
<td>IPv6_ADDRESS</td>
<td>A valid IPv6 address (with or without the prefix), such as fdd1:0:1:107::500.</td>
</tr>
<tr>
<td>IPv6_GATEWAY</td>
<td>A valid IPv6 gateway address, such as 2001:470:8418:280d::1.</td>
</tr>
</tbody>
</table>

**API request (1 IPv4 static route):**

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl1" -X "POST" -d "id=43463&set_routes=10.10.25.0|255.255.255.0|10.10.25.255|Route1"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

**API request (mix of IPv4 and IPv6 static routes):**

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl1" -X "POST" -d "id=43463&set_routes=192.168.0.0|255.255.255.0|10.100.11.157|Name2,192.168.10.0|255.255.255.0|10.100.11.157|Name3,192.167.0.0|255.255.255.0|10.100.11.157|Name4,fdd1:0:1:107::500|2001:470:8418:280d::1,||Name1,fdd1:0:1:107::500/64|2001:470:8418:280d::1"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
 <!DOCTYPE SIMPLE_RETURN SYSTEM
  "https://qualysapi.qualys.com/api/2.0/simple.return.dtd">
 <SIMPLE_RETURN>
  <RESPONSE>
```
<DATETIME>2014-07-09T08:49:18Z</DATETIME>

Virtual scanner updated successfully

<Item_list>
  <item>
    <id>43463</id>
  </item>
</item_list>

View Scanner Appliances with VLANs, Static Routes

Use the parameters "action=list" and "output_mode=full".

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "GET"
  "https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=list&id=43463&output_mode=full"
```

XML output:

```
...<vlans>
  <setting>Enabled</setting>
  <vlan>
    <id>0</id>
    <name>vlan1</name>
    <ip_address>10.10.10.1</ip_address>
    <netmask>255.255.255.0</netmask>
  </vlan>
</vlans>
<static_routes>
  <route>
    <name>Route1</name>
    <ip_address>10.10.25.0</ip_address>
    <netmask>255.255.255.0</netmask>
    <gateway>10.10.25.255</gateway>
  </route>
  <route>
    <name>Route2</name>
    <ip_address>10.10.26.0</ip_address>
    <netmask>255.255.255.0</netmask>
    <gateway>10.10.26.255</gateway>
  </route>
</static_routes>
...
Delete All VLAN Records
Use the "set_vlans" parameters and set it to " (empty string).

API request (deletes all VLAN records):
```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: -d "id=43463&set_vlans=" "https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-07-09T08:49:18Z</DATETIME>
    <TEXT>Virtual scanner updated successfully</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Delete All Static Route Records
Use the "set_routes" parameters and set it to " (empty string).

API request (deletes all static route records):
```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: -d "id=43463&set_routes=" "https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-07-09T08:49:18Z</DATETIME>
    <TEXT>Virtual scanner updated successfully</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```
Option Profile Export

/api/2.0/fo/subscription/option_profile/?action=export

[GET]

Export one option profile or all option profiles in the subscription to an XML file. Manager user role is required.
Permissions - The API user must have the Manager role.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=export</td>
<td>(Required)</td>
</tr>
<tr>
<td>output_format={XML}</td>
<td>(Optional) XML format is supported. When unspecified, output format is XML.</td>
</tr>
<tr>
<td>option_profile_id={value}</td>
<td>(Optional) By default all option profiles will be exported. Specify an option profile ID and we'll export the option profile matching this ID only.</td>
</tr>
<tr>
<td>option_profile_title={value}</td>
<td>(Optional) By default all option profiles will be exported. Specify a title and we'll export the option profile matching this title only - exact match is required.</td>
</tr>
<tr>
<td>option_profile_type={value}</td>
<td>(Optional) Option profile group name/type, e.g. user (for user defined), compliance (for compliance profile), pci (for PCI vulnerabilities profile). Note: &quot;option_profile_type&quot; parameter can be specified with &quot;option_profile_id&quot; or &quot;option_profile_title&quot;.</td>
</tr>
<tr>
<td>include_system_option_profiles ={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

DTD

<platform API server>/api/2.0/fo/subscription/option_profile/option_profile_info.dtd

Sample - Export Option Profiles
All the option profiles in the user’s account get exported in XML format.

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X GET "action=export" "https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/

XML response:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE OPTION_PROFILES SYSTEM
  "https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/option_profile_info.dtd">
<OPTION_PROFILES>
  <OPTION_PROFILE>
    <BASIC_INFO>
      <ID>111186</ID>
      <GROUP_NAME><![CDATA[OP-SCAN]]></GROUP_NAME>
      <GROUP_TYPE>user</GROUP_TYPE>
      <USER_ID><![CDATA[John Doe(john_doe)]]]></USER_ID>
      <UNIT_ID>0</UNIT_ID>
      <SUBSCRIPTION_ID>44</SUBSCRIPTION_ID>
      <IS_DEFAULT>0</IS_DEFAULT>
      <IS_GLOBAL>1</IS_GLOBAL>
      <IS_OFFLINE.Syncable>0</IS_OFFLINE.Syncable>
      <UPDATE_DATE>N/A</UPDATE_DATE>
    </BASIC_INFO>
    <SCAN>
      <TCP_PORTS>
        <TCP_PORTS_TYPE>full</TCP_PORTS_TYPE>
        <THREE_WAY_HANDSHAKE>1</THREE_WAY_HANDSHAKE>
      </TCP_PORTS>
      <UDP_PORTS>
        <UDP_PORTS_TYPE>none</UDP_PORTS_TYPE>
        <HAS_ADDITIONAL>1</HAS_ADDITIONAL>
        <ADDITIONAL_PORTS>1-1024,8080,8181</ADDITIONAL_PORTS>
      </UDP_PORTS>
      <AUTHORITATIVE_OPTION>1</AUTHORITATIVE_OPTION>
    </SCAN>
    <SCAN_DEAD_HOSTS>1</SCAN_DEAD_HOSTS>
    <CLOSE_VULNERABILITIES>
      <HAS_CLOSE_VULNERABILITIES>1</HAS_CLOSE_VULNERABILITIES>
      <HOST_NOT_FOUND_ALIVE>7</HOST_NOT_FOUND_ALIVE>
    </CLOSE_VULNERABILITIES>
    <PURGE_OLD_HOST_OS_CHANGED>1</PURGE_OLD_HOST_OS_CHANGED>
    <PERFORMANCE>
      <PARALLEL_SCALING>1</PARALLEL_SCALING>
      <OVERALL_PERFORMANCE>Custom</OVERALL_PERFORMANCE>
    </PERFORMANCE>
    <EXTERNAL_SCANNERS>30</EXTERNAL_SCANNERS>
    <SCANNER_APPLIANCES>48</SCANNER_APPLIANCES>
  </OPTION_PROFILE>
</OPTION_PROFILES>
<PROCESSES_TO_RUN>
<HTTP_PROCESSES>18</HTTP_PROCESSES>
</PROCESSES_TO_RUN>

<PACKET_DELAY>Minimum</PACKET_DELAY>

<PORT_SCANNING_AND_HOST_DISCOVERY>Minimum</PORT_SCANNING_AND_HOST_DISCOVERY>

</PERFORMANCE>

<LOAD_BALANCER_DETECTION>1</LOAD_BALANCER_DETECTION>

<PASSWORD_BRUTE_FORCING>

<SYSTEM>

<HAS_SYSTEM>1</HAS_SYSTEM>

<SYSTEM_LEVEL>Standard</SYSTEM_LEVEL>
</SYSTEM>

<CUSTOM_LIST>

<CUSTOM>

<ID>3001</ID>

<TITLE><![CDATA[123]]></TITLE>

<TYPE>FTP</TYPE>

<LOGIN_PASSWORD><![CDATA[L:temp,P:123123123]]></LOGIN_PASSWORD>
</CUSTOM>
</CUSTOM_LIST>

</PASSWORD_BRUTE_FORCING>

<VULNERABILITY_DETECTION>

<CUSTOM_LIST>

<CUSTOM>

<ID>2094</ID>

<TITLE><![CDATA[Option Profile: Qualys Top 20 Options]]></TITLE>
</CUSTOM>

<CUSTOM>

<ID>2095</ID>

<TITLE><![CDATA[Option Profile: 2008 SANS20 Options]]></TITLE>
</CUSTOM>

<CUSTOM>

<ID>2096</ID>

<TITLE><![CDATA[Scan Report Template: High Severity Report]]></TITLE>
</CUSTOM>

<CUSTOM>

<ID>5230</ID>

<TITLE><![CDATA[118960]]></TITLE>
</CUSTOM>

<CUSTOM>

<ID>87936</ID>

<TITLE><![CDATA[Bash Shellshock Detection]]></TITLE>
</CUSTOM>
</CUSTOM_LIST>
</VULNERABILITY_DETECTION>
<CUSTOM>
</CUSTOM>

<CUSTOM>
<ID>87937</ID>
<TITLE><![CDATA[Heartbleed Detection]]></TITLE>
</CUSTOM>

<CUSTOM>
<ID>87938</ID>
<TITLE><![CDATA[Windows Authentication Results v.1]]></TITLE>
</CUSTOM>

<CUSTOM>
<ID>87939</ID>
<TITLE><![CDATA[Unix Authentication Results v.1]]></TITLE>
</CUSTOM>

<CUSTOM>
<ID>87940</ID>
<TITLE><![CDATA[Inventory Results v.1]]></TITLE>
</CUSTOM>

<CUSTOM>
<ID>87941</ID>
<TITLE><![CDATA[SSL Certificates]]></TITLE>
</CUSTOM>

</CUSTOM_LIST>

<DETECTION_INCLUDE>
<BASIC_HOST_INFO_CHECKS>1</BASIC_HOST_INFO_CHECKS>
<OVAL_CHECKS>1</OVAL_CHECKS>
</DETECTION_INCLUDE>

<DETECTION_EXCLUDE>
</DETECTION_EXCLUDE>

</VULNERABILITY_DETECTION>

<AUTHENTICATION><![CDATA[Windows, Unix, Oracle, Oracle Listener, SNMP, VMware, DB2, HTTP, MySQL, Sybase]]></AUTHENTICATION>

<AUTHENTICATION_LEAST_PRIVILEGE><![CDATA[Unix]]></AUTHENTICATION_LEAST_PRIVILEGE>

<ADDL_CERT_DETECTION>1</ADDL_CERT_DETECTION>

<DISSOLVABLE_AGENT_ENABLE>1</DISSOLVABLE_AGENT_ENABLE>

<WINDOWS_SHARE_ENUMERATION_ENABLE>1</WINDOWS_SHARE_ENUMERATION_ENABLE>

</DISSOLVABLE_AGENT>

<LITE_OS_SCAN>1</LITE_OS_SCAN>
<CUSTOM_HTTP_HEADER>
<VALUE>AFCD</VALUE>
</CUSTOM_HTTP_HEADER>

<FILE_INTEGRITY_MONITORING>
<AUTO_UPDATEEXPECTED_VALUE>1</AUTO_UPDATEEXPECTED_VALUE>
</FILE_INTEGRITY_MONITORING>

<DO_NOT_OVERWRITE_OS>1</DO_NOT_OVERWRITE_OS>

<SYSTEM_AUTH_RECORD>
<INCLUDE_SYSTEM_AUTH>
<ON_DUPLICATE_USE_USER_AUTH>1</ON_DUPLICATE_USE_USER_AUTH>
</INCLUDE_SYSTEM_AUTH>
</SYSTEM_AUTH_RECORD>
</SCAN>

<MAP>
<BASIC_INFO_GATHERING_ON>netblockonly</BASIC_INFO_GATHERING_ON>
<TCP_PORTS>
<TCP_PORTS_STANDARD_SCAN>1</TCP_PORTS_STANDARD_SCAN>
<TCP_PORTS_ADDITIONAL>
HAS_ADDITIONAL>1</HAS_ADDITIONAL>
ADDITIONAL_PORTS>1,2,3,80</ADDITIONAL_PORTS>
</TCP_PORTS_ADDITIONAL>
</TCP_PORTS>
<UDP_PORTS>
<UDP_PORTS_STANDARD_SCAN>1</UDP_PORTS_STANDARD_SCAN>
<UDP_PORTS_ADDITIONAL>
HAS_ADDITIONAL>1</HAS_ADDITIONAL>
ADDITIONAL_PORTS>4,5,6,8181</ADDITIONAL_PORTS>
</UDP_PORTS_ADDITIONAL>
</UDP_PORTS>

<MAP_OPTIONS>
<PERFORM_LIVE_HOST_SWEEP>1</PERFORM_LIVE_HOST_SWEEP>
<DISABLE_DNS_TRAFFIC>1</DISABLE_DNS_TRAFFIC>
</MAP_OPTIONS>

<MAP_PERFORMANCE>
<OVERALL_PERFORMANCE>Custom</OVERALL_PERFORMANCE>
</MAP>

<MAP_PARALLEL>
<EXTERNAL_SCANNERS>16</EXTERNAL_SCANNERS>
<SCANNER_APPLIANCES>14</SCANNER_APPLIANCES>
</MAP_PARALLEL>

<PACKET_DELAY>Maximum</PACKET_DELAY>
</MAP_PERFORMANCE>

<MAP_AUTHENTICATION>VMware</MAP_AUTHENTICATION>
</MAP>

<ADDITIONAL>
<HOST_DISCOVERY>
<TCP_PORTS>
<STANDARD_SCAN>1</STANDARD_SCAN>
<TCP_ADDITIONAL>
<HAS_ADDITIONAL>1</HAS_ADDITIONAL>
<ADDITIONAL_PORTS>1-6,1024</ADDITIONAL_PORTS>
</TCP_ADDITIONAL>
</TCP_PORTS>
<UDP_PORTS>
<STANDARD_SCAN>1</STANDARD_SCAN>
</UDP_PORTS>
<ICMP>1</ICMP>
</HOST_DISCOVERY>
<BLOCK_RESOURCES>
<WATCHGUARD_DEFAULT_BLOCKED_PORTS>1</WATCHGUARD_DEFAULT_BLOCKED_PORTS>
<ALL_REGISTERED_IPS>1</ALL_REGISTERED_IPS>
</BLOCK_RESOURCES>
<PACKET_OPTIONS>
<IGNORE_FIREWALL_GENERATED_TCP_RST>1</IGNORE_FIREWALL_GENERATED_TCP_RST>
<IGNORE_ALL_TCP_RST>1</IGNORE_ALL_TCP_RST>
<IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>1</IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>
<NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>1</NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>
</PACKET_OPTIONS>
</ADDITIONAL>
</OPTION_PROFILE>
</OPTION_PROFILES>

Sample - Export Option Profile with specific title and ID

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl"
-X GET "action=export&option_profile_title=OP-COMP&option_profile_id=111235"
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/"

XML response:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE OPTION_PROFILES SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile_info.dtd">
<OPTION_PROFILES>
<OPTION_PROFILE>
  <BASIC_INFO>
    <ID>111235</ID>
    <GROUP_NAME><![CDATA[OP-COMP]]></GROUP_NAME>
    <GROUP_TYPE>compliance</GROUP_TYPE>
    <USER_ID><![CDATA[John Doe (john_doe)]]></USER_ID>
    <UNIT_ID>0</UNIT_ID>
    <SUBSCRIPTION_ID>44</SUBSCRIPTION_ID>
    <IS_GLOBAL>0</IS_GLOBAL>
    <UPDATE_DATE>N/A</UPDATE_DATE>
  </BASIC_INFO>
  <SCAN>
    <TARGETED_SCAN>1</TARGETED_SCAN>
  </SCAN>
  <PORTS>
    <TARGETED_SCAN>1</TARGETED_SCAN>
  </PORTS>
  <PERFORMANCE>
    <PARALLEL_SCALING>0</PARALLEL_SCALING>
    <OVERALL_PERFORMANCE>Normal</OVERALL_PERFORMANCE>
    <HOSTS_TO_SCAN>
      <EXTERNAL_SCANNERS>5</EXTERNAL_SCANNERS>
      <SCANNER_APPLIANCES>30</SCANNER_APPLIANCES>
    </HOSTS_TO_SCAN>
    <PROCESSES_TO_RUN>
      <TOTAL_PROCESSES>10</TOTAL_PROCESSES>
      <HTTP_PROCESSES>10</HTTP_PROCESSES>
    </PROCESSES_TO_RUN>
    <PACKET_DELAY>Short</PACKET_DELAY>
    <PORT_SCANNING_AND_HOST_DISCOVERY>Minimum</PORT_SCANNING_AND_HOST_DISCOVERY>
  </PERFORMANCE>
  <DISSOLVABLE_AGENT>
    <DISSOLVABLE_AGENT_ENABLE>1</DISSOLVABLE_AGENT_ENABLE>
    <PASSWORD_AUDITING_ENABLE>
      <HAS_PASSWORD_AUDITING_ENABLE>1</HAS_PASSWORD_AUDITING_ENABLE>
      <CUSTOM_PASSWORD_DICTIONARY>asdf</CUSTOM_PASSWORD_DICTIONARY>
    </PASSWORD_AUDITING_ENABLE>
    <WINDOWS_SHARE_ENUMERATION_ENABLE>1</WINDOWS_SHARE_ENUMERATION_ENABLE>
    <WINDOWS_DIRECTORY_SEARCH_ENABLE>1</WINDOWS_DIRECTORY_SEARCH_ENABLE>
  </DISSOLVABLE_AGENT>
  <CONTROL_TYPES>
    <FIM_CONTROLS_ENABLED>1</FIM_CONTROLS_ENABLED>
    <CUSTOM_WMI_QUERY_CHECKS>1</CUSTOM_WMI_QUERY_CHECKS>
  </CONTROL_TYPES>
  <TEST_AUTHENTICATION>1</TEST_AUTHENTICATION>
</OPTION_PROFILE>
<SYSTEM_AUTH_RECORD>
<INCLUDE_SYSTEM_AUTH>
<ON_DUPLICATE_USE_USER_AUTH>1</ON_DUPLICATE_USE_USER_AUTH>
</INCLUDE_SYSTEM_AUTH>
</SYSTEM_AUTH_RECORD>
</SCAN>

<ADDITIONAL>

<HOST_DISCOVERY>
<TCP_PORTS>
<STANDARD_SCAN>1</STANDARD_SCAN>
</TCP_PORTS>
<UDP_PORTS>
<STANDARD_SCAN>1</STANDARD_SCAN>
</UDP_PORTS>
<ICMP>1</ICMP>
</HOST_DISCOVERY>

<BLOCK_RESOURCES>
<WATCHGUARD_DEFAULT_BLOCKED_PORTS>1</WATCHGUARD_DEFAULT_BLOCKED_PORTS>
<ALL_REGISTERED_IPS>1</ALL_REGISTERED_IPS>
</BLOCK_RESOURCES>

<PACKET_OPTIONS>
<IGNORE_FIREWALL_GENERATED_TCP_RST>1</IGNORE_FIREWALL_GENERATED_TCP_RST>
<IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>1</IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>
<NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>1</NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>
</PACKET_OPTIONS>
</ADDITIONAL>

</OPTION_PROFILE>
</OPTION_PROFILES>

Sample - Export Option Profile with the specific title

API request:

curl --location
'qualys_base_url/api/2.0/fo/subscription/option_profile/?action=export&option_profile_title=VM_API_Option_profile_01' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'X-Requested-With: curl demo2' \
--header 'Accept: */*' \
--header 'Content-Length: 0' \
--header 'Authorization: Basic <encoded username:password string>''
XML response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE OPTION_PROFILES SYSTEM "<qualys_base_url>/api/2.0/fo/subscription/option_profile/option_profile_info.dtd">
<OPTION_PROFILES>
  <OPTION_PROFILE>
    <BASIC_INFO>
      <ID>2445050</ID>
      <GROUP_NAME>
        <![CDATA[VM_API_Option_profile_01]]>
      </GROUP_NAME>
      <GROUP_TYPE>user</GROUP_TYPE>
      <USER_ID>
        <![CDATA[Automation Test (scan_at)]]>
      </USER_ID>
      <UNIT_ID>0</UNIT_ID>
      <SUBSCRIPTION_ID>590924</SUBSCRIPTION_ID>
      <IS_DEFAULT>0</IS_DEFAULT>
      <IS_GLOBAL>0</IS_GLOBAL>
      <IS_OFFLINE_SYNCABLE>0</IS_OFFLINE_SYNCABLE>
      <UPDATE_DATE>2023-08-31T09:34:28Z</UPDATE_DATE>
    </BASIC_INFO>
    <SCAN>
      <PORTS>
        <TCP_PORTS>
          <TCP_PORTS_TYPE>none</TCP_PORTS_TYPE>
          <THREE_WAY_HANDSHAKE>0</THREE_WAY_HANDSHAKE>
        </TCP_PORTS>
        <UDP_PORTS>
          <UDP_PORTS_TYPE>none</UDP_PORTS_TYPE>
        </UDP_PORTS>
        <AUTHORITATIVE_OPTION>0</AUTHORITATIVE_OPTION>
      </PORTS>
      <SCAN_DEAD_HOSTS>0</SCAN_DEAD_HOSTS>
    </SCAN>
    <PURGE_OLD_HOST_OS_CHANGED>0</PURGE_OLD_HOST_OS_CHANGED>
    <PERFORMANCE>
      <!-- Other performance settings -->
    </PERFORMANCE>
  </OPTION_PROFILE>
</OPTION_PROFILES>
```
<TOTAL PROCESSES>10</TOTAL PROCESSES>
<HTTP PROCESSES>10</HTTP PROCESSES>
</PROCESSES TO RUN>
<PACKET_DELAY>Medium</PACKET_DELAY>

<PORT SCANNING AND HOST_DISCOVERY>Normal</PORT SCANNING_AND_HOST_DISCOVERY>
  <HOST CGI_CHECKS>0</HOST CGI_CHECKS>
  <MAX TARGETS PER SLICE>0</MAX TARGETS_PER_SLICE>
</PERFORMANCE>

<CONF SCAN LIMITED CONNECTIVITY>0</CONF SCAN LIMITED_CONNECTIVITY>
  <SKIP PRE SCANNING>0</SKIP PRE_SCANNING>
</PERFORMANCE>

<LOAD_BALANCER_DETECTION>0</LOAD_BALANCER_DETECTION>
</VULNERABILITY_DETECTION>
  <![](CDATA[complete])>
</DETECTION_INCLUDE>

<BASIC_HOST_INFO_CHECKS>0</BASIC_HOST_INFO_CHECKS>
  <OVAL_CHECKS>0</OVAL_CHECKS>
  <QRDI_CHECKS>0</QRDI_CHECKS>
</DETECTION_INCLUDE>
</VULNERABILITY_DETECTION>

<ADDL CERT DETECTION>0</ADDL CERT DETECTION>

<PERFORM_PARTIAL_SSL_TLS_AUDITING>1</PERFORM_PARTIAL_SSL_TLS_AUDITING>
</SCAN>

<MAP>
  <BASIC INFO GATHERING ON>none</BASIC_INFO_GATHERING_ON>
  <MAP_OPTIONS>
</MAP>

<PERFORM LIVE HOST SWEEP>0</PERFORM LIVE_HOST_SWEEP>
  <DISABLE DNS TRAFFIC>0</DISABLE DNS TRAFFIC>
</MAP_OPTIONS>

<OVERALL PERFORMANCE>Normal</OVERALL_PERFORMANCE>
  <MAP PARALLEL>
  <EXTERNAL SCANNERS>6</EXTERNAL_SCANNERS>
  <SCANNER APPLIANCES>8</SCANNER_APPLIANCES>
  <NETBLOCK SIZE>16384 IPs</NETBLOCK_SIZE>
</MAP_PARALLEL>
</MAP_PERFORMANCE>
Sample - Export Option Profile of type PCI
The option profile with PCI type in the user’s account get exported in XML format.

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X GET "action=export&option_profile_type=pci" "https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/"
```

XML response:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE OPTION_PROFILES SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/info.dtd">
<OPTION_PROFILES>
  <OPTION_PROFILE>
    <BASIC_INFO>
      <ID>111223</ID>
      <GROUP_NAME><![CDATA[PCI-Example]]></GROUP_NAME>
    </BASIC_INFO>
  </OPTION_PROFILE>
</OPTION_PROFILES>
```
<GROUP_TYPE>pci</GROUP_TYPE>
<User_ID><![CDATA[John Doe (john_doe)]></USER_ID>
<Unit_ID>0</Unit_ID>
<Subscription_ID>44</Subscription_ID>
<Is_Global>1</Is_Global>
<Is_Offline_Syncable>0</Is_Offline_Syncable>
<Update_Date>N/A</Update_Date>
</Basic_Info>
<Scan>
<Scan_Dead_Hosts>1</Scan_Dead_Hosts>
<Close_Vulnerabilities>
  <Has_Close_Vulnerabilities>1</Has_Close_Vulnerabilities>
  <Host_Not_Found_Alive>4</Host_Not_Found_Alive>
</Close_Vulnerabilities>
<Purge_Old_Host_OS_Changed>1</Purge_Old_Host_OS_Changed>
<Performance>
  <Parallel_Scaling>1</Parallel_Scaling>
  <Overall_Performance>Low</Overall_Performance>
  <Hosts_To_Scan>
    <External_Scanners>5</External_Scanners>
    <Scanner_Appliances>10</Scanner_Appliances>
  </Hosts_To_Scan>
  <Processes_To_Run>
    <Total_Processes>4</Total_Processes>
    <Http_Processes>2</Http_Processes>
  </Processes_To_Run>
  <Packet_Delay>Long</Packet_Delay>
</Performance>
</Scan>
<Additional>
<Host_Discovery>
  <Tcp_Ports>
    <Standard_Scan>1</Standard_Scan>
    <Tcp_Additional>
      <Has_Additional>1</Has_Additional>
      <Additional_Ports>1-6,1024</Additional_Ports>
    </Tcp_Additional>
  </Tcp_Ports>
</Host_Discovery>
</Additional>
</Option_Profile>
</Option_Profiles>
Sample - Export Options Profile for Database UDC

Export the Option Profile for MS SQL, Oracle, Sybase, PostgreSQL/Pivotal Greenplum, and IBM DB2 with database preference key setting and its corresponding value defined.

API request:
```
curl -u "username:password" -H "X-Requested-With:curl" -H "Content-type: text/xml" -X -d 
"action=export&option_profile_id=1710150"
"https://qualysapi.qualys.com/api/2.0/fo/sub{}scription/option_profile/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OPTION_PROFILES SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/subscrip{}tion/option_profile/option_profile_info.dtd">
<OPTION_PROFILES>
  <OPTION_PROFILE>
    ...
  </OPTION_PROFILE>
</OPTION_PROFILES>
<DATABASE_PREFERENCE_KEY>
  <MSSQL>
    <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
    <DB_UDC_LIMIT>250</DB_UDC_LIMIT>
  </MSSQL>
  <ORACLE>
    <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
    <DB_UDC_LIMIT>10</DB_UDC_LIMIT>
  </ORACLE>
  <SYBASE>
    <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
    <DB_UDC_LIMIT>60</DB_UDC_LIMIT>
  </SYBASE>
  <POSTGRESQL>
    <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
    <DB_UDC_LIMIT>2500</DB_UDC_LIMIT>
  </POSTGRESQL>
  <DB2>
    <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
    <DB_UDC_LIMIT>350</DB_UDC_LIMIT>
  </DB2>
</DATABASE_PREFERENCE_KEY>
<FILE_INTEGRITY_MONITORING>
  <AUTO_UPDATE_EXPECTED_VALUE>0</AUTO_UPDATEEXPECTED_VALUE>
```
Option Profile Import

/api/2.0/fo/subscription/option_profile/?action=import

[POST]

Import all option profiles defined in input XML file.

Permissions - The API user must have the Manager role.

When calling the Import Option Profile API the user needs to pass the proper XML with Content-Type XML. This will create option profiles in that user’s subscription. All validations are applied as in the Qualys portal UI while creating option profiles using the Import Option Profile API.

Validations and Constraints:

1) The Option Profile DTD is used to validate a generated/exported Option Profile XML file.

2) An XSD file is used to validate a proper format and required elements of the option profile XML file when importing this file.

3) While importing, any Search Lists defined for Vulnerability Detection, Custom and/or Excluded Lists, must be created in the user’s subscription before making an Import Option Profile call. At import time we try to match the Search List “title” to a search list title in the user’s subscription. If a match is found the search list is used, otherwise “Complete” Vulnerability Detection is assigned.

4) Password Brute Force Lists are not imported and will always be empty assigned, regardless of Option Profile XML content.

5) Policies defined for the PC Scan Restriction feature are not imported and will be empty assigned, regardless of Option Profile XML content.

Input Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=import</td>
<td>(Required)</td>
</tr>
</tbody>
</table>

Sample - Import option profiles in the input file into the user’s account

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -H "content-type: text/xml" -X POST --data-binary @Export_OP.xml "https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/?action=import"
Note: "Export_OP.xml" contains the request POST data.

Request POST data:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OPTION_PROFILES SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/subcription/option_profile/option_profile_info.dtd">
<OPTION_PROFILES>
    <OPTION_PROFILE>
        <BASIC_INFO>
            <ID>11123</ID>
            <GROUP_NAME><![CDATA[OP-SCAN]]></GROUP_NAME>
            <GROUP_TYPE>user</GROUP_TYPE>
            <USER_ID><![CDATA[John Doe (john_doe)]></USER_ID>
            <UNIT_ID>0</UNIT_ID>
            <SUBSCRIPTION_ID>76084</SUBSCRIPTION_ID>
            <IS_DEFAULT>0</IS_DEFAULT>
            <IS_GLOBAL>1</IS_GLOBAL>
            <IS_OFFLINE_SYNCABLE>0</IS_OFFLINE_SYNCABLE>
            <UPDATE_DATE>N/A</UPDATE_DATE>
        </BASIC_INFO>
        <SCAN>
            <TCP_PORTS>
                <TCP_PORTS_TYPE>full</TCP_PORTS_TYPE>
                <THREE_WAY_HANDSHAKE>1</THREE_WAY_HANDSHAKE>
            </TCP_PORTS>
            <UDP_PORTS>
                <UDP_PORTS_TYPE>none</UDP_PORTS_TYPE>
                <HAS_ADDITIONAL>1</HAS_ADDITIONAL>
                <ADDITIONAL_PORTS>1-1024,8080,8181</ADDITIONAL_PORTS>
            </UDP_PORTS>
            <AUTHORITATIVE_OPTION>1</AUTHORITATIVE_OPTION>
        </SCAN>
        <SCAN_DEAD_HOSTS>1</SCAN_DEAD_HOSTS>
        <CLOSE_VULNERABILITIES>
            <HAS_CLOSE_VULNERABILITIES>1</HAS_CLOSE_VULNERABILITIES>
            <HOST_NOT_FOUND_ALIVE>7</HOST_NOT_FOUND_ALIVE>
        </CLOSE_VULNERABILITIES>
        <PURGE_OLD_HOST_OS_CHANGED>1</PURGE_OLD_HOST_OS_CHANGED>
        <PERFORMANCE>
            <PARALLEL_SCALING>1</PARALLEL_SCALING>
            <OVERALL_PERFORMANCE>Custom</OVERALL_PERFORMANCE>
            <HOSTS_TO_SCAN>
```

161
<EXTERNAL_SCANNERS>30</EXTERNAL_SCANNERS>
<SCANNER_APPLIANCES>48</SCANNER_APPLIANCES>
</HOSTS_TO_SCAN>
<PROCESSES_TO_RUN>
<TOTAL_PROCESSES>18</TOTAL_PROCESSES>
<HTTP_PROCESSES>18</HTTP_PROCESSES>
</PROCESSES_TO_RUN>
<PACKET_DELAY>Maximum</PACKET_DELAY>
</PORT_SCANNING_AND_HOST_DISCOVERY>
</LOAD_BALANCER_DETECTION>
</PASSWORD_BRUTE_FORCING>
</SYSTEM>
<CUSTOM_LIST>
<CUSTOM>
<ID>3001</ID>
<TITLE><![CDATA[123]]></TITLE>
<TYPE>FTP</TYPE>
<LOGIN_PASSWORD><![CDATA[L:temp,P:123123123]]></LOGIN_PASSWORD>
</CUSTOM>
</CUSTOM_LIST>
</PASSWORD_BRUTE_FORCING>
</VULNERABILITY_DETECTION>
</CUSTOM_LIST>
</CUSTOM>
<ID>2094</ID>
<TITLE><![CDATA[Option Profile: Qualys Top 20 Options]]></TITLE>
</CUSTOM>
</CUSTOM>
<ID>2095</ID>
<TITLE><![CDATA[Option Profile: 2008 SANS20 Options]]></TITLE>
</CUSTOM>
</CUSTOM>
<ID>2096</ID>
<TITLE><![CDATA[Scan Report Template: High Severity Report]]></TITLE>
</CUSTOM>
<ID>5230</ID>
<TITLE><![CDATA[118960]]></TITLE>
</CUSTOM>

</CUSTOM>

</CUSTOM>

<TITLE><![CDATA[Bash Shellshock Detection]]></TITLE>
</CUSTOM>

<TITLE><![CDATA[Heartbleed Detection]]></TITLE>
</CUSTOM>

<TITLE><![CDATA[Windows Authentication Results v.1]]></TITLE>
</CUSTOM>

<TITLE><![CDATA[Unix Authentication Results v.1]]></TITLE>
</CUSTOM>

<TITLE><![CDATA[Inventory Results v.1]]></TITLE>
</CUSTOM>

<TITLE><![CDATA[SSL Certificates]]></TITLE>
</CUSTOM>

</CUSTOM_LIST>
<DETECTION_INCLUDE>

<BASIC_HOST_INFO_CHECKS>1</BASIC_HOST_INFO_CHECKS>
</DETECTION_INCLUDE>

</CUSTOM_LIST>
<DETECTION_EXCLUDE>

<CUSTOM_LIST>

<ID>2099</ID>
<TITLE><![CDATA[DL]]></TITLE>
</CUSTOM>

</CUSTOM_LIST>
</DETECTION_EXCLUDE>
</VULNERABILITY_DETECTION>

<AUTHENTICATION><![CDATA[Windows,Unix,Oracle,Oracle Listener,SNMP,VMware,DB2,HTTP,MySQL,Sybase]]></AUTHENTICATION>

<ADDL_CERT_DETECTION>1</ADDL_CERT_DETECTION>

<DISSOLVABLE_AGENT>

<DISSOLVABLE_AGENT_ENABLE>1</DISSOLVABLE_AGENT_ENABLE>
<WINDOWS_SHARE_ENUMERATION_ENABLE>1</WINDOWS_SHARE_ENUMERATION_ENABLE>

</DISSOLVABLE_AGENT>
<LITE_OS_SCAN>1</LITE_OS_SCAN>
<CUSTOM_HTTP_HEADER>
  <VALUE>AFCD</VALUE>
</CUSTOM_HTTP_HEADER>

<FILE_INTEGRITY_MONITORING>
  <AUTO_UPDATEEXPECTED_VALUE>1</AUTO_UPDATEEXPECTED_VALUE>
</FILE_INTEGRITY_MONITORING>
<DO_NOT_OVERWRITE_OS>1</DO_NOT_OVERWRITE_OS>

<SYSTEM_AUTH_RECORD>
  <INCLUDE_SYSTEM_AUTH>
    <ONDUPLICATE_USE_USER_AUTH>1</ONDUPLICATE_USE_USER_AUTH>
  </INCLUDE_SYSTEM_AUTH>
</SYSTEM_AUTH_RECORD>

</SCAN>

<MAP>

<BASIC_INFO_GATHERING_ON>netblockonly</BASIC_INFO_GATHERING_ON>
<TCP_PORTS>
  <TCP_PORTS_STANDARD_SCAN>1</TCP_PORTS_STANDARD_SCAN>
  <TCP_PORTS_ADDITIONAL>
    <HAS_ADDITIONAL>1</HAS_ADDITIONAL>
    <ADDITIONAL_PORTS>1,2,3,80</ADDITIONAL_PORTS>
  </TCP_PORTS_ADDITIONAL>
</TCP_PORTS>

<UDP_PORTS>
  <UDP_PORTS_STANDARD_SCAN>1</UDP_PORTS_STANDARD_SCAN>
  <UDP_PORTS_ADDITIONAL>
    <HAS_ADDITIONAL>1</HAS_ADDITIONAL>
    <ADDITIONAL_PORTS>4,5,6,8181</ADDITIONAL_PORTS>
  </UDP_PORTS_ADDITIONAL>
</UDP_PORTS>

<MAP_OPTIONS>
  <PERFORM_LIVE_HOST_SWEEP>1</PERFORM_LIVE_HOST_SWEEP>
  <DISABLE_DNS_TRAFFIC>1</DISABLE_DNS_TRAFFIC>
</MAP_OPTIONS>

<MAP_PERFORMANCE>
  <OVERALL_PERFORMANCE>Custom</OVERALL_PERFORMANCE>
  <MAP_PARALLEL>
    <EXTERNAL_SCANNERS>16</EXTERNAL_SCANNERS>
    <SCANNER_APPLIANCES>14</SCANNER_APPLIANCES>
    <NETBLOCK_SIZE>64</NETBLOCK_SIZE>
  </MAP_PARALLEL>
  <PACKET_DELAY>Medium</PACKET_DELAY>
</MAP_PERFORMANCE>
<MAP_AUTHENTICATION>VMware</MAP_AUTHENTICATION>
</MAP>
<ADDITIONAL>

<HOST_DISCOVERY>
<TCP_PORTS>
  <STANDARD.Scan>1</STANDARD.Scan>
  <TCP_ADDITIONAL>
    <HAS_ADDITIONAL>1</HAS_ADDITIONAL>
    <ADDITIONAL_PORTS>1-6,1024</ADDITIONAL_PORTS>
  </TCP_ADDITIONAL>
</TCP_PORTS>
<UDP_PORTS>
  <STANDARD.Scan>1</STANDARD.Scan>
</UDP_PORTS>
<ICMP>1</ICMP>
</HOST_DISCOVERY>

<BLOCK_RESOURCES>
<WATCHGUARD_DEFAULT_BLOCKED_PORTS>1</WATCHGUARD_DEFAULT_BLOCKED_PORTS>
  <ALL_REGISTERED_IPS>1</ALL_REGISTERED_IPS>
</BLOCK_RESOURCES>

<PACKET_OPTIONS>
<IGNORE_FIREWALL_GENERATED_TCP_RST>1</IGNORE_FIREWALL_GENERATED_TCP_RST>
  <IGNORE_ALL_TCP_RST>1</IGNORE_ALL_TCP_RST>
<IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>1</IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>
  <NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>1</NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>
</PACKET_OPTIONS>
</ADDITIONAL>
</OPTION_PROFILE>
</OPTION_PROFILES>

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2017-04-03T11:17:43Z</DATETIME>
    <TEXT>Successfully imported Option profile for the subscription Id 76084</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
Sample - Import option profiles

API request:

```
curl --location
'<qualys_base_url>/api/2.0/fo/subscription/option_profile/?action=import'
  
  --header 'X-Requested-With: curl demo2'
  --header 'Accept: */**'
  --header 'Content-Type: text/xml'
  --header 'Authorization: Basic <encoded username:password string>'
  
  --data '<?xml version="1.0" encoding="UTF-8" ?>
```

Request POST data:

```
<!DOCTYPE OPTION_PROFILES SYSTEM
"<qualys_base_url>/api/2.0/fo/subscription/option_profile/option_profile_info.dtd">

<OPTION_PROFILES>
  <OPTION_PROFILE>
    <BASIC_INFO>
      <ID>1599154</ID>
      <GROUP_NAME><![CDATA[API_IMPORT_OP_1647433780]]></GROUP_NAME>
      <GROUP_TYPE>user</GROUP_TYPE>
      <USER_ID><![CDATA[Network Disabled (pv_nd)]></USER_ID>
      <UNIT_ID>0</UNIT_ID>
      <SUBSCRIPTION_ID>915185</SUBSCRIPTION_ID>
      <IS_DEFAULT>0</IS_DEFAULT>
      <IS_GLOBAL>0</IS_GLOBAL>
      <IS_OFFLINE_SYNCABLE>0</IS_OFFLINE_SYNCABLE>
      <UPDATE_DATE>2019-03-19T09:13:26Z</UPDATE_DATE>
    </BASIC_INFO>
    <SCAN>
      <PORTS>
        <TCP_PORTS>
          <TCP_PORTS_TYPE>standard</TCP_PORTS_TYPE>
          <THREE_WAY_HANDSHAKE>0</THREE_WAY_HANDSHAKE>
        </TCP_PORTS>
      </PORTS>
    </SCAN>
  </OPTION_PROFILE>
</OPTION_PROFILES>
```
<TCP_PORTS>
<UDP_PORTS>
<UDP_PORTS_TYPE>standard</UDP_PORTS_TYPE>
</UDP_PORTS>
<AUTHORITATIVE_OPTION>0</AUTHORITATIVE_OPTION>
</PORTS>
<SCAN_DEAD_HOSTS>0</SCAN_DEAD_HOSTS>
<PERFORMANCE>
<PARALLEL_SCALING>0</PARALLEL_SCALING>
<OVERALL_PERFORMANCE>Normal</OVERALL_PERFORMANCE>
<HOSTS_TO_SCAN>
<EXTERNAL沈ANNERS>15</EXTERNAL沈ANNERS>
<SCANNER_APlIANCES>30</SCANNER_APlIANCES>
</HOSTS_TO_SCAN>
<PROCESSES_TO_RUN>
<TOTAL_PROCESSES>10</TOTAL_PROCESSES>
<HTTP_PROCESSES>10</HTTP_PROCESSES>
</PROCESSES_TO_RUN>
<PACKET_DELAY>Medium</PACKET_DELAY>
</PORT_SCANNING_AND_HOST_DISCOVER>
<LOAD_BALANCER_DETECTION>0</LOAD_BALANCER_DETECTION>
<VULNERABILITY_DETECTION>
<COMPLETE><![CDATA[complete]]></COMPLETE>
<DETECTION_INCLUDE>
<BASIC_HOST_INFO_CHECKS>0</BASIC_HOST_INFO_CHECKS>
<OVAL_CHECKS>0</OVAL_CHECKS>
</DETECTION_INCLUDE>
</VULNERABILITY_DETECTION>
<AUTHENTICATION><![CDATA[Windows, Unix, Oracle, Oracle Listener, SNMP, VMware, DB2, HTTP, MySQL, MongoDB, Tomcat Server, Oracle Weblogic Server, Palo Alto Networks Firewall, Jboss Server]]></AUTHENTICATION>
<ADDL_CERT_DETECTION>0</ADDL_CERT_DETECTION>
<TEST_AUTHENTICATION>1</TEST_AUTHENTICATION>
<PERFORM_PARTIAL_SSL_TLS_AUDITING>1</PERFORM_PARTIAL_SSL_TLS_AUDITING>
</SCAN>
<MAP>
<BASIC_INFO_GATHERING_ON>all</BASIC_INFO_GATHERING_ON>
<TCP_PORTS>
<TCP_PORTS_STANDARD.Scan>1</TCP_PORTS_STANDARD.Scan>
</TCP_PORTS>
<MAP_OPTIONS>
<PERFORM_LIVE_HOST_Sweep>1</PERFORM_LIVE_HOST_Sweep>
<DISABLE_DNS_TRAFFIC>0</DISABLE_DNS_TRAFFIC>
</MAP_OPTIONS>
<MAP_PERFORMANCE>
<OVERALL_PERFORMANCE>Normal</OVERALL_PERFORMANCE>
<MAP_PARALLEL>
<EXTERNAL_SCANNERS>6</EXTERNAL_SCANNERS>
<SCANNER_APPLIANCES>8</SCANNER_APPLIANCES>
<NETBLOCK_SIZE>16384 IP-s</NETBLOCK_SIZE>
</MAP_PARALLEL>
<PACKET_DELAY>Minimum</PACKET_DELAY>
</MAP_PERFORMANCE>
<MAP_AUTHENTICATION>none</MAP_AUTHENTICATION>
</MAP>
<ADDITIONAL>
<HOST_DISCOVERY>
<TCP_PORTS>
<STANDARD.Scan>1</STANDARD.Scan>
</TCP_PORTS>
<UDP_PORTS>
<STANDARD.Scan>1</STANDARD.Scan>
</UDP_PORTS>
<ICMP>1</ICMP>
</HOST_DISCOVERY>
<PACKET_OPTIONS>
<IGNORE_FIREWALL_GENERATED_TCP_RST>0</IGNORE_FIREWALL_GENERATED_TCP_RST>
<IGNORE_ALL_TCP_RST>0</IGNORE_ALL_TCP_RST>
<IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>0</IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>
<NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>0</NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>
</PACKET_OPTIONS>
</ADDITIONAL>
</OPTION_PROFILE>
</OPTION_PROFILES>
XML Response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "<qualys_base_url>/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2023-08-31T10:08:47Z</DATETIME>
    <TEXT>Successfully imported Option profile for the subscription Id 590924</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>2445059</KEY>
        <VALUE>API_IMPORT_OP_1647433780</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Option Profiles for VM

/api/2.0/fo/subscription/option_profile/vm/

Create, update, list and delete option profiles for VM scans.

Permissions - All users will be able to list option profiles. A Manager will be able to create, update, and delete option profiles in the subscription, and a Unit Manager will be able to create, update, and delete option profiles for users in their business unit.

Create VM Option Profile

/api/2.0/fo/subscription/option_profile/vm/?action=create

[POST]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=create</td>
<td>(Required)</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required) A title for easy identification.</td>
</tr>
<tr>
<td>owner={value}</td>
<td>(Optional) The owner of the option profile(s), or the user who created the option profile.</td>
</tr>
<tr>
<td>default={0</td>
<td>1}</td>
</tr>
<tr>
<td>enable_partial_ssl_tls_auditing={0</td>
<td>1}</td>
</tr>
<tr>
<td>global={0</td>
<td>1}</td>
</tr>
<tr>
<td>offline_scanner={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_tcp_ports={none</td>
<td>full</td>
</tr>
<tr>
<td>scan_tcp_ports_additional={port1.port2}</td>
<td>(Optional) Specify additional ports to scan (up to 12500 ports).</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3_way_handshake={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_udp_ports={none</td>
<td>full</td>
</tr>
<tr>
<td>vulnerability_detection={complete</td>
<td>custom</td>
</tr>
<tr>
<td>scan_udp_ports_additional={port1,port2}</td>
<td>(Optional) Specify additional ports to scan (up to 20500 ports).</td>
</tr>
<tr>
<td>authoritative_option={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_dead_hosts={0</td>
<td>1}</td>
</tr>
<tr>
<td>close_vuln_on_dead_hosts={0</td>
<td>1}</td>
</tr>
<tr>
<td>not_found_alive_times={value}</td>
<td>(Optional) Specify the number of times the host is not found alive after which the vulnerability should be closed. This setting is available only when close_vuln_on_dead_hosts=1.</td>
</tr>
<tr>
<td>purge_host_data={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
### Scan Configuration

**Option Profiles for VM**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>external_scanners_use={value}</code></td>
<td>(Optional) Specify the maximum number of external scanners to use for scanning perimeter assets. (This option is available when your subscription is configured with multiple external scanners).</td>
</tr>
<tr>
<td>`scan_parallel_scaling={0</td>
<td>1}`</td>
</tr>
<tr>
<td>`scan_overall_performance={high</td>
<td>normal</td>
</tr>
<tr>
<td><code>scan_external_scanners={value}</code></td>
<td>(Optional) Specify the number of external scanners to be used for associated scans. This setting is available only if you have multiple external scanners in your subscription. For example, if you have 10 external scanners in your subscription, you can configure this setting to any number between 1 to 10.</td>
</tr>
<tr>
<td><code>scan_scanner_appliances={value}</code></td>
<td>(Optional) Specify the number of scanner appliances to scan at the same time (per scan task). Launching several concurrent scans on the same scanner appliance has a multiplying effect on bandwidth usage and may exceed available scanner resources. Don’t have scanner appliances? Disregard the Scanner Appliance setting.</td>
</tr>
<tr>
<td><code>scan_total_process={value}</code></td>
<td>(Optional) Specify the maximum number of processes to run at the same time per host. Note that the total number of processes includes the HTTP processes.</td>
</tr>
<tr>
<td><code>scan_http_process={value}</code></td>
<td>(Optional) Specify the maximum number of HTTP processes to run at the same time.</td>
</tr>
<tr>
<td>`scan_packet_delay={minimum</td>
<td>short</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>scan_intensity={normal</td>
<td>medium</td>
</tr>
<tr>
<td>load_balancer={0</td>
<td>1}</td>
</tr>
<tr>
<td>password_brute_forcing_system={minimal</td>
<td>limited</td>
</tr>
<tr>
<td>password_brute_forcing_custom={value1,value2}</td>
<td>(Optional) Specify titles of the login/password pairs you create for password brute forcing on the Qualys Cloud Platform UI.</td>
</tr>
<tr>
<td>custom_search_list_ids={value1,value2}</td>
<td>(Optional) Specify ids of search lists you want to use in your scan.</td>
</tr>
<tr>
<td>custom_search_list_title={value1,value2}</td>
<td>(Optional) Specify titles of search lists you want to use in your scan.</td>
</tr>
<tr>
<td>basic_host_information_checks={0</td>
<td>1}</td>
</tr>
<tr>
<td>oval_checks={0</td>
<td>1}</td>
</tr>
<tr>
<td>all_qrdi_checks={0</td>
<td>1}</td>
</tr>
<tr>
<td>exclude_search_list_ids={value1,value2}</td>
<td>(Optional) Specify ids of search lists you want to exclude from your scan.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>authentication={value1, value2}</td>
<td>(Optional) Want to run authenticated scans? When you use authentication we’ll perform a more in-depth assessment and get you the most accurate results with fewer false positives. Specify one or more technologies for the hosts you want to scan. Be sure you’ve configured authentication records (under Scans &gt; Authentication) before running your scan. The following options are available: - Windows - Unix - Oracle - Oracle Listener - SNMP - VMware - DB2 - HTTP - MySQL - MongoDB - Tomcat Server - Palo Alto Networks Firewall - Sybase</td>
</tr>
<tr>
<td>authentication_least_privilege=Unix</td>
<td>(Optional) Specify authentication_least_privilege=Unix (this value is case sensitive) to use the least privileges required for Unix authentication. When specified, the scanner will not pass root delegation information specified in the Unix record to the scanner for vulnerability scans. When not specified (the default), root delegation will be used if specified in the Unix record. Note: Unix authentication must be enabled in the same option profile (authentication=Unix).</td>
</tr>
<tr>
<td>enable_additional_certificate_detection={0</td>
<td>1}</td>
</tr>
<tr>
<td>enable_dissolvable_agent={0</td>
<td>1}</td>
</tr>
<tr>
<td>enable_windows_share Enumeration={0</td>
<td>1}</td>
</tr>
<tr>
<td>enable_lite_os_scan={0</td>
<td>1}</td>
</tr>
<tr>
<td>custom_http_header={value}</td>
<td>(Optional) Specify a custom value in order to drop defenses (such as logging, IPs, etc) when authorized scans are being run.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>custom_http_definition_key={value}</td>
<td>(Optional) Specify a custom HTTP header definition key</td>
</tr>
<tr>
<td>custom_http_definition_header={value}</td>
<td>(Optional) Specify a value for the custom HTTP header definition key defined in custom_http_definition_key.</td>
</tr>
<tr>
<td>host_alive_testing={0</td>
<td>1}</td>
</tr>
<tr>
<td>not_overwrite_os={0</td>
<td>1}</td>
</tr>
<tr>
<td>test_authentication={0</td>
<td>1}</td>
</tr>
<tr>
<td>enable_max_scan_duration_per_asset={0</td>
<td>1}</td>
</tr>
<tr>
<td>max_scan_duration_per_asset_minutes=maximum</td>
<td>Maximum duration in minutes for scan to be performed on each asset. The parameters enable_max_scan_duration_per_asset and max_scan_duration_per_asset are mutually exclusive, and can only be specified if enable_max_scan_duration_per_asset is 1.</td>
</tr>
</tbody>
</table>

### System Authentication

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_system_auth={0</td>
<td>1}</td>
</tr>
<tr>
<td>use_system_auth_on_duplicate={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>use_user_auth_on_duplicate={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

### Map

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic_information_gathering=[all</td>
<td>register</td>
</tr>
<tr>
<td>map_tcp_ports_standard_scan={0</td>
<td>1}</td>
</tr>
<tr>
<td>map_tcp_ports_additional={value1,value2}</td>
<td>(Optional) Specify additional TCP ports to scan. You can specify up to 20 ports including the standard scan ports.</td>
</tr>
<tr>
<td>map_udp_ports_standard_scan={0</td>
<td>1}</td>
</tr>
<tr>
<td>map_udp_ports_additional={value1,value2}</td>
<td>(Optional) Specify additional UDP ports to scan. You can specify up to 10 ports including the standard scan ports.</td>
</tr>
<tr>
<td>perform_live_host_sweep={0</td>
<td>1}</td>
</tr>
<tr>
<td>disable_dns_traffic={0</td>
<td>1}</td>
</tr>
<tr>
<td>map_overall_performance={high</td>
<td>normal</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>map_external_scanners={value}</td>
<td>(Optional) Specify the number of external scanners for netblocks to map at the same time per scanner. This setting is available only if you have multiple external scanners in your subscription. For example, if you have 10 external scanners in your subscription, you can configure this setting to any number between 1 to 10.</td>
</tr>
<tr>
<td>map_scanner_appliances={value}</td>
<td>(Optional) Specify the number of scanner appliances for netblocks to map at the same time per scanner. Launching several concurrent scans on the same scanner appliance has a multiplying effect on bandwidth usage and may exceed available scanner resources. Don’t have scanner appliances? Disregard the Scanner Appliance setting.</td>
</tr>
<tr>
<td>map_netblock_size={1024 IPs</td>
<td>4096 IPs</td>
</tr>
<tr>
<td>map_packet_delay={minimum</td>
<td>short</td>
</tr>
<tr>
<td>map_authentication={VMware}</td>
<td>(Optional) Authentication enables the scanner to log into hosts at scan time to extend detection capabilities. See the online help to learn how to configure this option.</td>
</tr>
</tbody>
</table>

**Additional**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>additional_tcp_ports={0</td>
<td>1}</td>
</tr>
<tr>
<td>additional_tcp_ports_standard_scan={0</td>
<td>1}</td>
</tr>
<tr>
<td>additional_tcp_ports_additional={value1,value2}</td>
<td>(Optional) Specify additional TCP ports to scan. You can specify up to 20 ports including the standard scan ports.</td>
</tr>
<tr>
<td>additional_udp_ports={0</td>
<td>1}</td>
</tr>
<tr>
<td>additional_udp_ports_type={standard</td>
<td>custom}</td>
</tr>
<tr>
<td>additional_udp_ports_custom={value1,value2}</td>
<td>(Optional) Specify additional UDP ports to scan. You can specify up to 10 ports including the standard scan ports.</td>
</tr>
<tr>
<td>icmp={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
Scan Configuration
Option Profiles for VM

Parameter | Description
---|---
blocked_resources=[0|1] | (Optional) Specify 1 in order to add ports protected by your firewall/IDS to prevent them from being scanned.
protected_ports=[default|custom] | (Optional) Ports protected by your firewall/IDS. Specify "default" to provide a list of default blocked ports: 0-1, 111, 513-514, 2049, 4100, 6000-6005, 7100, 8000. Default setting is "default". Specify "custom" to provide a custom list of protected ports using protected_ports_custom.
protected_ports_custom=[value1,value2] | (Optional) Specify a custom list of protected ports.
protected_ips=[all|custom] | (Optional) IP addresses and ranges protected by your firewall/IDS. Default is "all".
protected_ips_custom=[value1,value2] | (Optional) Specify a custom list of IP addresses and ranges protected by your firewall/IDS.
ignore_firewall_generated_tcp_rst_packets=[0|1] | (Optional) Specify 1 to identify firewall-generated TCP RESET packets and ignore them.
ignore_allTcp_rst_packets=[0|1] | (Optional) Specify 1 to ignore all TCP RESET packets - firewall-generated and live-host-generated.
ignore_firewall_generated_tcp_syn_ack_packets=[0|1] | (Optional) Specify 1 to determine if TCP SYN-ACK packets are generated by a filtering device and ignore packets that appear to originate from such devices.
not_send_tcp_ack_or_syn_ack_packets_during_host_discovery=[0|1] | (Optional) Specify 1 if you do not want to send TCP ACK or SYN-ACK packets. Out of state TCP packets are not SYN packets and do not belong to an existing TCP session.

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST 
"/api/2.0/fo/subscription/option_profile/vm/
"action=create&title=99&global=1&scan_tcp_ports=full&scan_udp_ports=standard&scan_overall_performance=normal&vulnerability_detection=all"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM 
"http://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
 <RESPONSE>
  <DATETIME>2018-04-26T06:40:03Z</DATETIME>
  <TEXT>Option profile successfully added.</TEXT>
  <ITEM_LIST>
   <ITEM>
    <KEY>ID</KEY>
   </ITEM>
  </ITEM_LIST>
 </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Create VM Option Profile with SSL/TLS auditing enabled.

API Request:
```
curl --location --request POST
```

XML Response:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "qualys_base_url/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2023-08-31T10:05:16Z</DATETIME>
    <TEXT>Option profile successfully added.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>2445054</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Update VM Option Profile

/api/2.0/fo/subscription/option_profile/vm/?action=update

[POST]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update (Required)</td>
<td>(Required) The ID of the option profile.</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Optional) Use to enable or disable the partial SSL/TLS auditing during scan execution. Specify 1 to enable partial SSL/TLS checks while executing the scan.</td>
</tr>
<tr>
<td>enable_partial_ssl_tls_auditing = {0</td>
<td>1}</td>
</tr>
</tbody>
</table>

For a list of optional parameters, see Input Parameters for Create VM Option Profile.

API request:


XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "http://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-26T09:51:15Z</DATETIME>
    <TEXT>Option profile successfully updated.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>   
        <VALUE>25121</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Update VM option profile with SSL/TLS auditing disabled

API Request:

curl --location --request POST "qualys_base_url/api/2.0/fo/subscription/option_profile/vm/?enable_partial_ssl_tls_auditing=0&%20action=update&id=2437618&scan_tcp_ports=standar
d&scan_udp_ports=standard&vulnerability_detection=runtime' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'X-Requested-With: curl demo2' \
--header 'Accept: */*' \
--header 'Content-Length: 0' \
--header 'Authorization: Basic <encoded username:password string>'

XML Response:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "<qualys_base_url>/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
 <RESPONSE>
  <DATETIME>2023-08-31T10:04:24Z</DATETIME>
  <TEXT>Option profile successfully updated.</TEXT>
  <ITEM_LIST>
   <ITEM>
    <KEY>ID</KEY>
    <VALUE>2437618</VALUE>
   </ITEM>
  </ITEM_LIST>
 </RESPONSE>
</SIMPLE_RETURN>
VM Option Profile List

/api/2.0/fo/subscription/option_profile/vm/?action=list

[GET] [POST]

Input Parameters

All option profiles are fetched if no parameters are given. To fetch a specific option profile, provide the “id” or “title” parameter with the option profile id or title of interest. Optionally, you can filter the results by using optional parameters listed under Input Parameters for Create VM Option Profile.

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X GET
"action=list"
"http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/vm/"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE OPTION_PROFILES SYSTEM
"http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/option_profile_info.dtd">
<OPTION_PROFILES>
<OPTION_PROFILE>
<BASIC_INFO>
<ID>51451401</ID>
<GROUP_NAME><![CDATA[user op - 1]]></GROUP_NAME>
<GROUP_TYPE>user</GROUP_TYPE>
<USER_ID><![CDATA[John smith (jsmith_ap)]></USER_ID>
<UNIT_ID>0</UNIT_ID>
<SUBSCRIPTION_ID>10421401</SUBSCRIPTION_ID>
<IS_DEFAULT>0</IS_DEFAULT>
<IS_GLOBAL>1</IS_GLOBAL>
<IS_OFFLINE_SYNCABLE>1</IS_OFFLINE_SYNCABLE>
<UPDATE_DATE>2018-04-10T13:39:41Z</UPDATE_DATE>
</BASIC_INFO>
<SCAN>
<PORTS>
<TCP_PORTS>
<TCP_PORTS_TYPE>standard</TCP_PORTS_TYPE>
<TCP_PORTS_ADDITIONAL>
<HAS_ADDITIONAL>1</HAS_ADDITIONAL>
<ADDITIONAL_PORTS>1024</ADDITIONAL_PORTS>
</TCP_PORTS_ADDITIONAL>
<THREE_WAY_HANDSHAKE>1</THREE_WAY_HANDSHAKE>
</TCP_PORTS>
<UDP_PORTS>
   <UDP_PORTS_TYPE>light</UDP_PORTS_TYPE>
   <UDP_PORTS_ADDITIONAL>
      <HAS_ADDITIONAL>1</HAS_ADDITIONAL>
      <ADDITIONAL_PORTS>8080</ADDITIONAL_PORTS>
   </UDP_PORTS_ADDITIONAL>
</UDP_PORTS>
<AUTHORITATIVE_OPTION>1</AUTHORITATIVE_OPTION>
</PORTS>
<SCAN_DEAD_HOSTS>1</SCAN_DEAD_HOSTS>
<CLOSE_VULNERABILITIES>
   <HAS_CLOSE_VULNERABILITIES>1</HAS_CLOSE_VULNERABILITIES>
   <HOST_NOT_FOUND_ALIVE>10</HOST_NOT_FOUND_ALIVE>
</CLOSE_VULNERABILITIES>
<PURGE_OLD_HOST_OS_CHANGED>1</PURGE_OLD_HOST_OS_CHANGED>
<PERFORMANCE>
   <PARALLEL_SCALING>1</PARALLEL_SCALING>
   <OVERALL_PERFORMANCE>Normal</OVERALL_PERFORMANCE>
   <HOSTS_TO_SCAN>
      <EXTERNAL_SCANNERS>10</EXTERNAL_SCANNERS>
      <SCANNER_APPLIANCES>30</SCANNER_APPLIANCES>
   </HOSTS_TO_SCAN>
   <PROCESSES_TO_RUN>
      <TOTAL_PROCESSES>10</TOTAL_PROCESSES>
      <HTTP_PROCESSES>10</HTTP_PROCESSES>
   </PROCESSES_TO_RUN>
   <PACKET_DELAY>Medium</PACKET_DELAY>
</PERFORMANCE>
<LOAD_BALANCER_DETECTION>1</LOAD_BALANCER_DETECTION>
<PASSWORD_BRUTE_FORCING>
   <SYSTEM>
      <HAS_SYSTEM>1</HAS_SYSTEM>
      <SYSTEM_LEVEL>Standard</SYSTEM_LEVEL>
   </SYSTEM>
</PASSWORD_BRUTE_FORCING>
<CUSTOM_LIST>
   <CUSTOM>
      <ID>1001</ID>
      <TITLE><![CDATA[ftp - 1]]></TITLE>
      <TYPE>FTP</TYPE>
   </CUSTOM>
</CUSTOM_LIST>
<LOGIN_PASSWORD><![CDATA[L:Guest,P:temp]]></LOGIN_PASSWORD>
</CUSTOM>
</CUSTOM>
<ID>1002</ID>
<TITLE><![CDATA[ssh - 1]]></TITLE>
<TYPE>SSH</TYPE>

<LOGIN_PASSWORD><![CDATA[L:Guest,P:temp]]></LOGIN_PASSWORD>
</CUSTOM>

<CUSTOM>
<ID>1003</ID>
<TITLE><![CDATA[window - 1]]></TITLE>
<TYPE>Windows</TYPE>

<LOGIN_PASSWORD><![CDATA[L:Guest,P:temp]]></LOGIN_PASSWORD>
</CUSTOM>
</CUSTOM_LIST>
</PASSWORD_BRUTE_FORCING>

<VULNERABILITY_DETECTION>
<COMPLETE><![CDATA[complete]]></COMPLETE>
<DETECTION_INCLUDE>
<BASIC_HOST_INFO_CHECKS>0</BASIC_HOST_INFO_CHECKS>
<OVAL_CHECKS>1</OVAL_CHECKS>
</DETECTION_INCLUDE>
</VULNERABILITY_DETECTION>

<AUTHENTICATION><![CDATA[Windows,Unix,Oracle,Oracle Listener,SNMP,VMware,DB2,HTTP,MySQL,Sybase]]></AUTHENTICATION>

<AUTHENTICATION_LEAST_PRIVILEGE><![CDATA[Unix]]></AUTHENTICATION_LEAST_PRIVILEGE>

<ADDL_CERT_DETECTION>1</ADDL_CERT_DETECTION>
<DISSOLVABLE_AGENT>
<DISSOLVABLE_AGENT_ENABLE>1</DISSOLVABLE_AGENT_ENABLE>

<WINDOWS_SHARE_ENUMERATION_ENABLE>1</WINDOWS_SHARE_ENUMERATION_ENABLE>

</DISSOLVABLE_AGENT>
<LITE_OS_SCAN>1</LITE_OS.Scan>

<CUSTOM_HTTP_HEADER>
<VALUE>sdfdfs</VALUE>
<DEFINITION_KEY>abc</DEFINITION_KEY>
<DEFINITION_VALUE>xyz</DEFINITION_VALUE>
</CUSTOM_HTTP_HEADER>

<System.Auth.Record>
<INCLUDE_SYSTEM_AUTH>
<ON_DUPLICATE_USE_USER_AUTH>1</ON_DUPLICATE_USE_USER_AUTH>
</INCLUDE_SYSTEM_AUTH>
</SYSTEM_AUTH_RECORD>
</SCAN>

</MAP>
<BASIC_INFO_GATHERING_ON>all</BASIC_INFO_GATHERING_ON>
<TCP_PORTS>
  <TCP_PORTS_STANDARD_SCAN>1</TCP_PORTS_STANDARD_SCAN>
  <TCP_PORTS_ADDITIONAL>
    <HAS_ADDITIONAL>1</HAS_ADDITIONAL>
    <ADDITIONAL_PORTS>2</ADDITIONAL_PORTS>
  </TCP_PORTS_ADDITIONAL>
</TCP_PORTS>
<UDP_PORTS>
  <UDP_PORTS_STANDARD_SCAN>1</UDP_PORTS_STANDARD_SCAN>
  <UDP_PORTS_ADDITIONAL>
    <HAS_ADDITIONAL>1</HAS_ADDITIONAL>
    <ADDITIONAL_PORTS>9</ADDITIONAL_PORTS>
  </UDP_PORTS_ADDITIONAL>
</UDP_PORTS>
<Map_Options>
  <PERFORM_LIVE_HOST_SWEEP>1</PERFORM_LIVE_HOST_SWEEP>
  <DISABLE_DNS_TRAFFIC>1</DISABLE_DNS_TRAFFIC>
</Map_Options>
<Map_Performance>
  <OVERALL_PERFORMANCE>Custom</OVERALL_PERFORMANCE>
  <MAP_PARALLEL>
    <EXTERNAL_SCANNERS>10</EXTERNAL_SCANNERS>
    <SCANNER_APPLIANCES>12</SCANNER_APPLIANCES>
    <NETBLOCK_SIZE>8192 IPs</NETBLOCK_SIZE>
  </MAP_PARALLEL>
  <PACKET_DELAY>Medium</PACKET_DELAY>
</Map_Performance>
<Map_Authentication>VMware</Map_Authentication>
</Map>
<Additional>
  <Host_Discovery>
    <TCP_PORTS>
      <STANDARD_SCAN>1</STANDARD_SCAN>
      <TCP_ADDITIONAL>
        <HAS_ADDITIONAL>1</HAS_ADDITIONAL>
        <ADDITIONAL_PORTS>1024</ADDITIONAL_PORTS>
      </TCP_ADDITIONAL>
    </TCP_PORTS>
    <UDP_PORTS>
      <CUSTOM_PORT><![CDATA[69,111]]></CUSTOM_PORT>
    </UDP_PORTS>
    <ICMP>1</ICMP>
  </Host_Discovery>
</Additional>
<Block_Resources>
<WATCHGUARD_DEFAULT_BLOCKED_PORTS>1</WATCHGUARD_DEFAULT_BLOCKED_PORTS>
   <ALL_REGISTERED_IPS>1</ALL_REGISTERED_IPS>
</BLOCK_RESOURCES>
<PACKET_OPTIONS>
   <IGNORE_FIREWALL_GENERATED_TCP_RST>1</IGNORE_FIREWALL_GENERATED_TCP_RST>
   <IGNORE_ALL_TCP_RST>1</IGNORE_ALL_TCP_RST>
   <IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>1</IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>
   <NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>1</NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>
</PACKET_OPTIONS>
</ADDITIONAL>
</OPTION_PROFILE>
</OPTION_PROFILES>

DTD
<platform API server>/api/2.0/fo/subscription/option_profile/option_profile_info.dtd

Delete VM Option Profile
/api/2.0/fo/subscription/option_profile/vm/?action=delete

[GET] [POST]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) The ID of the option profile.</td>
</tr>
</tbody>
</table>

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"action=delete&id=25121"
"http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/vm/"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"http://qualysapi.qualys.com/api/2.0/simple_return.dtd">

186
Option Profiles for PCI

/api/2.0/fo/subscription/option_profile/pci/

Create, update, list and delete option profiles for PCI.

Permissions - All users will be able to list option profiles. A Manager will be able to create, update, and delete option profiles in the subscription, and a Unit Manager will be able to create, update, and delete option profiles for users in their business unit.

Create PCI Option Profile

/api/2.0/fo/subscription/option_profile/pci/?action=create

[POST]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=create</td>
<td>(Required)</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required) A title for easy identification.</td>
</tr>
<tr>
<td>owner={value}</td>
<td>(Optional) The owner of the option profile(s), or the user who created the option profile.</td>
</tr>
<tr>
<td>global={0</td>
<td>1}</td>
</tr>
<tr>
<td>offline_scanner={0</td>
<td>1}</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>scan_parallel_scaling={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_overall_performance={high</td>
<td>normal</td>
</tr>
<tr>
<td>scan_external_scanners={value}</td>
<td>(Optional) Specify the number of external scanners to be used for associated scans. This setting is available only if you have multiple external scanners in your subscription. For example, if you have 10 external scanners in your subscription, you can configure this setting to any number between 1 to 10.</td>
</tr>
<tr>
<td>scan_scanner_appliances={value}</td>
<td>(Optional) Specify the number of scanner appliances to scan at the same time (per scan task). Launching several concurrent scans on the same scanner appliance has a multiplying effect on bandwidth usage and may exceed available scanner resources. Don’t have scanner appliances? Disregard the Scanner Appliance setting.</td>
</tr>
<tr>
<td>scan_total_process={value}</td>
<td>(Optional) Specify the maximum number of processes to run at the same time per host. Note that the total number of processes includes the HTTP processes.</td>
</tr>
<tr>
<td>scan_http_process={value}</td>
<td>(Optional) Specify the maximum number of HTTP processes to run at the same time.</td>
</tr>
<tr>
<td>scan_packet_delay={minimum</td>
<td>short</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`scan_intensity=[normal</td>
<td>medium</td>
</tr>
<tr>
<td>`scan_dead_hosts=[0</td>
<td>1]`</td>
</tr>
<tr>
<td>`close_vuln_on_dead_hosts=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>not_found_alive_times={value}</code></td>
<td><em>(Optional)</em> Specify the number of times the host is not found alive after which the vulnerability should be closed. This setting is available only when <code>close_vuln_on_dead_hosts=1</code>.</td>
</tr>
<tr>
<td>`purge_host_data=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>additional_tcp_ports_additional={value1,value2}</code></td>
<td><em>(Optional)</em> Specify additional TCP ports to scan. You can specify up to 7 additional ports apart from the 13 standard scan ports used by default: 21-23, 25, 53, 80, 88, 110-111, 135, 139, 443, 445.</td>
</tr>
</tbody>
</table>

### API request:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST "action=create&title=pci&global=1&offline_scanner=1&external_scanners_use=3&scan_parallel_scaling=1&scan_overall_performance=high&additional_tcp_ports_additional=80,35" "http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pci/"
```

### XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "http://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
```

189
Update PCI Option Profile

/api/2.0/fo/subscription/option_profile/pci/?action=update

[POST]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) The ID of the option profile.</td>
</tr>
</tbody>
</table>

For a list of optional parameters, see Input Parameters for Create PCI Option Profile.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"action=update&id=31102&title=pci2"
"http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pci/
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "http://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-10T10:32:50Z</DATETIME>
    <TEXT>Option profile successfully updated.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>31102</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
**PCI Option Profile List**

/`api/2.0/fo/subscription/option_profile/pci/?action=list`

[GET] [POST]

**Input Parameters**

All option profiles are fetched if no parameters are given. To fetch a specific option profile, provide the “id” or “title” parameter with the option profile id or title of interest. Optionally, you can filter the results by using optional parameters listed under Input Parameters for Create PCI Option Profile.

**API request:**

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X GET
  "action=list"
  "http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pci/"
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE OPTION_PROFILES SYSTEM
  "http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/option_profile_info.dtd">
<OPTION_PROFILES>
  <OPTION_PROFILE>
    <BASIC_INFO>
      <ID>31102</ID>
      <GROUP_NAME><![CDATA[jp pci 11]]></GROUP_NAME>
      <GROUP_TYPE>pci</GROUP_TYPE>
      <USER_ID><![CDATA[John Smith (jsmith_ap)]></USER_ID>
      <UNIT_ID>0</UNIT_ID>
      <SUBSCRIPTION_ID>10421401</SUBSCRIPTION_ID>
      <IS_GLOBAL>1</IS_GLOBAL>
      <IS_OFFLINE_SYNCABLE>0</IS_OFFLINE_SYNCABLE>
      <UPDATE_DATE>2018-04-10T10:32:50Z</UPDATE_DATE>
    </BASIC_INFO>
    <SCAN>
      <SCAN_DEAD_HOSTS>0</SCAN_DEAD_HOSTS>
      <PURGE_OLD_HOST_OS_CHANGED>0</PURGE_OLD_HOST_OS_CHANGED>
    </SCAN>
    <PERFORMANCE>
      <PARALLEL_SCALING>0</PARALLEL_SCALING>
      <OVERALL_PERFORMANCE>high</OVERALL_PERFORMANCE>
      <HOSTS_TO_SCAN>
        <EXTERNAL_SCANNERS>20</EXTERNAL_SCANNERS>
        <SCANNER_APPLIANCES>40</SCANNER_APPLIANCES>
      </HOSTS_TO_SCAN>
    </PERFORMANCE>
  </OPTION_PROFILE>
</OPTION_PROFILES>
```
<OPTION_PROFILE>
  <BASIC_INFO>
    <ID>51471401</ID>
    <GROUP_NAME><![CDATA[pci op - 1]]></GROUP_NAME>
    <GROUP_TYPE>pci</GROUP_TYPE>
    <USER_ID><![CDATA[John Smith (jsmith_ap)]></USER_ID>
    <UNIT_ID>0</UNIT_ID>
    <SUBSCRIPTION_ID>10421401</SUBSCRIPTION_ID>
    <IS_GLOBAL>0</IS_GLOBAL>
    <IS_OFFLINE_SYNCABLE>0</IS_OFFLINE_SYNCABLE>
    <UPDATE_DATE>2018-04-10T10:32:50Z</UPDATE_DATE>
  </BASIC_INFO>
  <SCAN>
    <SCAN_DEAD_HOSTS>1</SCAN_DEAD_HOSTS>
    <PURGE_OLD_HOST_OS_CHANGED>0</PURGE_OLD_HOST_OS_CHANGED>
    <PERFORMANCE>
      <PARALLEL_SCALING>1</PARALLEL_SCALING>
      <OVERALL_PERFORMANCE>High</OVERALL_PERFORMANCE>
      <HOSTS_TO_SCAN>
        <EXTERNAL_SCANNERS>20</EXTERNAL_SCANNERS>
        <SCANNER_APPLIANCES>40</SCANNER_APPLIANCES>
      </HOSTS_TO_SCAN>
      <PROCESSES_TO_RUN>
        <TOTAL PROCESSES>15</TOTAL PROCESSES>
        <HTTP PROCESSES>15</HTTP PROCESSES>
      </PROCESSES_TO_RUN>
      <PACKET_DELAY>Short</PACKET_DELAY>
      <PORT_SCANNING_AND_HOST_DISCOVERY>Normal</PORT_SCANNING_AND_HOST_DISCOVERY>
    </PERFORMANCE>
  </SCAN>
</OPTION_PROFILE>
Delete PCI Option Profile

/api/2.0/fo/subscription/option_profile/pci/?action=delete

[GET] [POST]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) The ID of the option profile.</td>
</tr>
</tbody>
</table>

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"action=delete&id=51471401"
"http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pci/

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"http://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-10T10:32:50Z</DATETIME>
    <TEXT>Option Profile Deleted Successfully</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
Option Profiles for Compliance

/api/2.0/fo/subscription/option_profile/pc/

Create, update, list and delete option profiles for compliance scans.

Permissions
Note: The list PC option profiles API is available as part of one of the following subscription combinations only:
- PC and API add-on
- PC, SCA, and API add-on
- VMDR, SCA, and API add-on

All users will be able to list option profiles. A Manager will be able to create, update, and delete option profiles in the subscription, and a Unit Manager will be able to create, update, and delete option profiles for users in their business unit.

Create PC Option Profile

@api/2.0/fo/subscription/option_profile/pc/?action=create

[POST]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=create</td>
<td>(Required)</td>
</tr>
<tr>
<td>title=[value]</td>
<td>(Required) The title for the option profile.</td>
</tr>
<tr>
<td>owner=[value]</td>
<td>(Optional) The owner of the option profile(s), or the user who created the option profile.</td>
</tr>
<tr>
<td>global=[0][1]</td>
<td>(Optional) Share this profile with other users by making it global. Are you a Manager? This profile will be available to all users. Are you a Unit Manager? This profile will be available to all users in your business unit. Specify 1 to make global.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scan_parallel_scaling={0</td>
<td>1}</td>
</tr>
<tr>
<td>scan_overall_performance={high</td>
<td>normal</td>
</tr>
<tr>
<td>scan_external_scanners={value}</td>
<td>(Optional) Specify the number of external scanners to be used for associated scans. This setting is available only if you have multiple external scanners in your subscription. For example, if you have 10 external scanners in your subscription, you can configure this setting to any number between 1 to 10.</td>
</tr>
<tr>
<td>scan_scanner_appliances={value}</td>
<td>(Optional) Specify the number of scanner appliances to scan at the same time (per scan task). Launching several concurrent scans on the same scanner appliance has a multiplying effect on bandwidth usage and may exceed available scanner resources. Don’t have scanner appliances? Disregard the Scanner Appliance setting.</td>
</tr>
<tr>
<td>scan_total_process={value}</td>
<td>(Optional) Specify the maximum number of processes to run at the same time per host. Note that the total number of processes includes the HTTP processes.</td>
</tr>
<tr>
<td>scan_http_process={value}</td>
<td>(Optional) Specify the maximum number of HTTP processes to run at the same time.</td>
</tr>
<tr>
<td>scan_packet_delay={minimum</td>
<td>short</td>
</tr>
</tbody>
</table>
### Parameter | Description
---|---
scan_intensity={normal|medium|low|minimum} | (Optional) This setting determines the aggressiveness (parallelism) of port scanning and host discovery at the port level. Lowering the intensity level has the effect of serializing port scanning and host discovery. This is useful for certain network conditions like cascading firewalls and lower scan prioritization on the network. Tip - If you are scanning through a firewall we recommended you reduce the intensity level. Unauthenticated scans see more of a performance difference using this option.
scan_by_policy={0|1} | (Optional) Specify 1 to enable scan by policy. The Scan by Policy option allows you to restrict your scans to the controls in specified policies. You can choose up to 20 policies, one policy at a time. Once you’ve specified a policy, all controls in that policy will be scanned including any special control types in the policy. This is regardless of the Control Types settings in the profile.
policy_names={value1,value2} | (Optional) Specify policy names to scan by policy.
policy_ids={value1,value2} | (Optional) Specify policy IDs to scan by policy.
auto_update_expected_value={0|1} | (Optional) Specify 1 to update the control expected value used for posture evaluation with the actual value returned by the scan.
fim_controls_enabled={0|1} | (Optional) Specify 1 to perform file integrity monitoring based on user defined file integrity checks. A file integrity check is a user defined control that checks for changes to a specific file. You should set auto_update_expected_value=1 in order to use this parameter.
custom_wmi_query_checks={0|1} | (Optional) Specify 1 to run Windows WMI query checks. When enabled, WMI query checks will be performed for user defined WMI Query Check controls.
enable_dissolvable_agent={0|1} | (Optional) Specify 1 to enable dissolvable agent. This is required for certain scan features like Windows Share Enumeration. How does it work? At scan time the Agent is installed on Windows devices to collect data, and once the scan is complete it removes itself completely from target systems.
enable_password_auditing={0|1} | (Optional) Specify 1 to check for service provided password auditing controls (control IDs 3893, 3894 and 3895). These controls are used to identify 1) user accounts with empty passwords, 2) user accounts with the password equal to the user name, and 3) user accounts with passwords equal to an entry in a user-defined password dictionary. This setting is available only if enable_dissolvable_agent=1.
custom_password_dictionary={value1,value2} | (Optional) Specify passwords in order to create a password dictionary. This is used when evaluating control ID 3895, which identifies user accounts where the password is equal to an entry in the password dictionary.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`enable_windows_share Enumeration=[0</td>
<td>1]`</td>
</tr>
<tr>
<td>`enable_windows_directory_search=[0</td>
<td>1]`</td>
</tr>
<tr>
<td>`scan_ports={standard</td>
<td>targeted}`</td>
</tr>
<tr>
<td>`mssql_db_udc_restriction=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>mssql_db_udc_limit={value}</code></td>
<td>(Optional) Provide a value to define the number of rows to be returned per scan (default is 256).</td>
</tr>
<tr>
<td>`oracle_db_udc_restriction=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>oracle_db_udc_limit={value}</code></td>
<td>(Optional) Provide a value to define the number of rows to be returned per scan (default is 5000).</td>
</tr>
<tr>
<td>`sybase_db_udc_restriction=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>sybase_db_udc_limit={value}</code></td>
<td>(Optional) Provide a value to define the number of rows to be returned per scan (default is 256). Maximum allowed limit for Sybase is 2500 rows.</td>
</tr>
<tr>
<td>`postgresql_db_udc_restriction=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>postgresql_db_udc_limit={value}</code></td>
<td>(Optional) Provide a value to define the number of rows to be returned per scan (default is 256). Maximum allowed limit for PostgreSQL/Pivotal Greenplum is 5000 rows.</td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
`sapiq_db_udc_restriction={0|1}` | (Optional) Set value to 1 if you want to specify a limit on the number of rows to be returned per scan for custom SAP IQ Database checks. If unspecified (sapiq_db_udc_restriction=0), we will auto discover instances at the installation directory level. This parameter can only be specified if `auto_auth_types=IBM WebSphere App Server`.

`sapiq_db_udc_limit={value}` | (Optional) Provide a value to define the number of rows to be returned per scan (default is 256). Maximum allowed limit for SAP IQ is 10000 rows.

`db2_db_udc_restriction={0|1}` | (Optional) Set value to 1 if you want to specify a limit on the number of rows to be returned per scan for custom IBM DB2 Database checks.

`db2_db_udc_limit={value}` | (Optional) The default value is 256 and maximum allowed limit is 5000 rows.

`enable_auth_instance_discovery={0|1}` | (Optional to create or update option profile record) Specify `enable_auth_instance_discovery=1` to enable auto discover instances and system record creation for the chosen auth types. When unspecified (enable_auth_instance_discovery=0), we will not scan to auto discover instances. The parameters `enable_auth_instance_discovery`, `scan_by_policy` and `include_system_auth` are mutually exclusive and cannot be specified together in the same request. In UI, this parameter is a check box and referred to "Allow instance discovery..." in the System Authentication Records section on the New/Edit Compliance Profile page.

`auto_auth_types={value}` | (Optional to create or update option profile record) Specify the technologies for which you want to enable auto discover instances and system record creation. The valid values are: Apache Web Server, IBM WebSphere App Server, Jboss Server, Tomcat Server, Oracle and MongoDB. Multiple technologies are specified as comma separated values. This parameter can only be specified if `enable_auth_instance_discovery=1`.

`ibm_was_discovery_mode={value}` | (Optional to create or update option profile record) Specify `ibm_was_discovery_mode=server_dir` to auto discover instances at the server directory level. Specify `ibm_was_discovery_mode=installation_dir` to auto discover instances at the installation directory level. When unspecified and `auto_auth_types=IBM WebSphere App Server`, we will auto discover instances at the installation directory level. This parameter can only be specified if `auto_auth_types` includes IBM WebSphere App Server.

`oracle_template_id={value}` | (Optional) The Template ID for the Oracle system record template you want to assign to the compliance profile for discovery scans. When `auto_auth_types=Oracle` is specified, then `oracle_template_id` or `oracle_template_name` must also be specified.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oracle_template_name=</td>
<td>(Optional) The Template Name for the Oracle system record template you want to assign to the compliance profile for discovery scans. When auto_auth_types=Oracle is specified, then oracle_template_id or oracle_template_name must also be specified.</td>
</tr>
<tr>
<td>mongodb_template_id=</td>
<td>(Optional) The Template ID for the MongoDB system record template you want to assign to the compliance profile for discovery scans. When auto_auth_types=MongoDB is specified, then mongodb_template_id or mongodb_template_name must also be specified.</td>
</tr>
<tr>
<td>mongodb_template_name=</td>
<td>(Optional) The Template Name for the MongoDB system record template you want to assign to the compliance profile for discovery scans. When auto_auth_types=MongoDB is specified, then mongodb_template_id or mongodb_template_name must also be specified.</td>
</tr>
<tr>
<td>include_system_auth=</td>
<td>(Optional to create or update option profile record) Specify include_system_auth=1 if you have a system created auth record and user created auth record for the same instance configuration and choose which one to include for scans. When unspecified (include_system_auth=0), user record will be selected for scan by default.</td>
</tr>
<tr>
<td></td>
<td>When include_system_auth=1, one of these parameters should be enabled: use_system_auth_on_duplicate or use_user_auth_on_duplicate.</td>
</tr>
<tr>
<td></td>
<td>In UI, this parameter is a check box and referred to &quot;Use System Authentication Records&quot; in the System Authentication Records section in the Scan tab on the New/Edit Compliance Profile page.</td>
</tr>
<tr>
<td>use_system_auth_on_duplicate=</td>
<td>(Optional to create or update option profile record) Specify use_system_auth_on_duplicate=1 to include system created auth record if you have a system record and user record for the same instance configuration. The parameters use_system_auth_on_duplicate and use_user_auth_on_duplicate are mutually exclusive and can only be specified if &quot;include_system_auth=1&quot;.</td>
</tr>
<tr>
<td>use_user_auth_on_duplicate=</td>
<td>(Optional to create or update option profile record) Specify use_user_auth_on_duplicate=1 to include user created authentication record if you have a system record and user record for the same instance. The parameters use_system_auth_on_duplicate and use_user_auth_on_duplicate are mutually exclusive and can only be specified if &quot;include_system_auth=1&quot;.</td>
</tr>
<tr>
<td>enable_instance_data_collection=</td>
<td>(Optional) Specify 1 to enable database instance data collection by using underlying OS authentication record. By default, this option is disabled.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>instance_data_collection_auth_types</td>
<td>(Optional) Specify the database technologies for which you want to enable OS authentication-based data collection. The valid values are: IBM DB2, InformixDB, MongoDB, MSSQL, MySQL, Oracle, Pivotal Greenplum, PostgreSQL, Sybase. You can use this parameter only if you set the value of the enable_instance_data_collection parameter to 1.</td>
</tr>
<tr>
<td>enable_os_based_instance_discovery=0</td>
<td>1]</td>
</tr>
<tr>
<td>os_based_instance_disc_technologies</td>
<td>(Optional) Specify a comma-separated list of technologies to enable OS authentication-based data collection. Currently we support Oracle JRE and IBM WebSphere Liberty. Hence, the valid values are: Oracle JRE and IBM WebSphere Liberty. You can use this parameter only if you set the value of the enable_os_based_instance_discovery parameter to 1.</td>
</tr>
<tr>
<td>Additional</td>
<td></td>
</tr>
<tr>
<td>additional_tcp_ports=0</td>
<td>1]</td>
</tr>
<tr>
<td>additional_tcp_ports_standard_scan=0</td>
<td>1]</td>
</tr>
<tr>
<td>additional_tcp_ports_additional=[value1,value2]</td>
<td>(Optional) Specify additional TCP ports to scan. You can specify up to 20 ports including the standard scan ports.</td>
</tr>
<tr>
<td>additional_udp_ports=0</td>
<td>1]</td>
</tr>
<tr>
<td>additional_udp_ports_type={standard</td>
<td>custom}</td>
</tr>
<tr>
<td>additional_udp_ports_custom=[value1,value2]</td>
<td>(Optional) Specify additional UDP ports to scan. You can specify up to 10 ports including the standard scan ports.</td>
</tr>
<tr>
<td>icmp=0</td>
<td>1]</td>
</tr>
<tr>
<td>blocked_resources=0</td>
<td>1]</td>
</tr>
<tr>
<td>protected_ports=default</td>
<td>custom]</td>
</tr>
<tr>
<td>protected_ports_custom=[value1,value2]</td>
<td>(Optional) Specify a custom list of protected ports.</td>
</tr>
</tbody>
</table>
Scan Configuration
Option Profiles for Compliance

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"action=create&title=pcjp&global=1&scan_parallel_scaling=1&scan_overall_performance=high&scan_by_policy=1&policy_names=jp2&auto_update_expected_value=1&scan_ports=standard&additional_tcp_ports=1&not_send_ack_or_syn_ack_packets_during_host_discovery=1"
"http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pc/
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"http://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-10T11:10:36Z</DATETIME>
    <TEXT>Compliance Option profile successfully added.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>39044</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected_ips=[all</td>
<td>custom]</td>
</tr>
<tr>
<td>protected_ips_custom={value1,value2}</td>
<td>(Optional) Specify a custom list of IP addresses and ranges protected by your firewall/IDS.</td>
</tr>
<tr>
<td>ignore_rst_packets={0</td>
<td>1}</td>
</tr>
<tr>
<td>ignore_firewall_generated_syn_ack_packets={0</td>
<td>1}</td>
</tr>
<tr>
<td>not_send_ack_or_syn_ack_packets_during_host_discovery={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
Sample create option profile for Oracle instance discovery

In this sample we are creating an option profile with instance discovery and system record creation enabled for Oracle and we’re using template ID 2237327.

**API request:**

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d "action=create&title=Profile-Auth-Ins-Oracle&enable_auth_instance_discovery=1&auto_auth_types=Oracle&scan_ports=targeted&oracle_template_id=2237327"
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pc/"

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2020-04-23T19:12:10Z</DATETIME>
    <TEXT>Compliance Option profile successfully added.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>3305478</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
**Sample create option profile for MongoDB instance discovery**

In this sample we are creating an option profile with the option to enable instance discovery and system record creation for MongoDB and we’re using template ID 6731346.

**API request:**
```bash
```

**XML output:**
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2022-12-19T07:48:27Z</DATETIME>
    <TEXT>Compliance Option profile successfully added.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>6863066</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

**Database UDCs for MS SQL, Oracle, Sybase, PostgreSQL/Pivotal Greenplum, SAP IQ, and IBM DB2**

We have added the following parameters to the Options Profile API to help you set a limit on the number of rows returned per scan for the MS SQL, Oracle, Sybase, PostgreSQL/Pivotal Greenplum, SAP IQ, and IBM DB2 UDCs.

- `DATABASE_PREFERENCE_KEY`
- `mssql_db_udc_restriction`
- `mssql_db_udc_limit`
- `oracle_db_udc_restriction`
- `oracle_db_udc_limit`
- `sybase_db_udc_restriction`
- sybase_db_udc_limit
- postgreSQL_db_udc_restriction
- postgreSQL_db_udc_limit
- sapiq_db_udc_restriction
- sapiq_db_udc_limit
- db2_db_udc_restriction
- db2_db_udc_limit

Maximum allowed limit for MS SQL is 256 rows, for Oracle, PostgreSQL/Pivotal Greenplum, and IBM DB2, it’s 5000 rows, for Sybase it’s 2500 rows, and for SAP IQ, it’s 10000 rows.

**Sample - Create for Database UDC**

Create with Database Preference Key and custom Limit set for MS SQL, Oracle, Sybase, PostgreSQL/Pivotal Greenplum, SAP IQ, and IBM DB2.

**API request:**

```bash
curl -u "username:password" -H "X-Requested-With:curl" -H "Content-type: text/xml" -X POST -d "action=create&type=api-pc-op&scan_ports=targeted&oracle_db_udc_restriction=1&oracle_db_udc_limit=10&msql_db_udc_restriction=1&msql_db_udc_limit=250&sybase_db_udc_restriction=1&sybase_db_udc_limit=50&postgreSQL_db_udc_restriction=1&postgreSQL_db_udc_limit=50&db2_db_udc_restriction=1&db2_db_udc_limit=300" "https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pc/
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2019-05-20T19:16:41Z</DATETIME>
    <TEXT>Compliance Option profile successfully added.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>1710286</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Update Compliance Option Profile
/api/2.0/fo/subscription/option_profile/pc/?action=update

[POST]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) The ID of the option profile.</td>
</tr>
</tbody>
</table>

For a list of optional parameters, see Input Parameters for Create PC Option Profile.

API request:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST "action=update&title=pc-jp&id=51491401" "http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pc/"
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "http://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-10T11:10:36Z</DATETIME>
    <TEXT>Compliance Option profile successfully updated.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>51491401</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Update for Database UDC
Update Option Profile with Oracle Database Preference Key

API request:

XML output:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2019-05-20T06:45:00Z</DATETIME>
    <TEXT>Compliance Option profile successfully updated.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>1709710</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
Compliance Option Profile List

/api/2.0/fo/subscription/option_profile/pc/?action=list

[GET] [POST]

Input Parameters

All option profiles are fetched if no parameters are given. To fetch a specific option profile, provide the “id” or “title” parameter with the option profile id or title of interest. Optionally, you can filter the results by using optional parameters listed under Input Parameters for Create PC Option Profile.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X GET
"action=list"
"http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pc/"
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE OPTION_PROFILES SYSTEM
"http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile_info.dtd">
<OPTION_PROFILES>
  <OPTION_PROFILE>
    <BASIC_INFO>
      <ID>19026</ID>
      <GROUP_NAME><![CDATA[Initial PC Options 2]]></GROUP_NAME>
      <GROUP_TYPE>compliance</GROUP_TYPE>
      <USER_ID><![CDATA[John Smith (jsmith_ap)]></USER_ID>
      <UNIT_ID>0</UNIT_ID>
      <SUBSCRIPTION_ID>10421401</SUBSCRIPTION_ID>
      <IS_GLOBAL>1</IS_GLOBAL>
      <UPDATE_DATE>2018-04-10T11:10:36Z</UPDATE_DATE>
    </BASIC_INFO>
    <SCAN>
      <PORTS>
        <TARGETED_SCAN>1</TARGETED_SCAN>
      </PORTS>
      <PERFORMANCE>
        <PARALLEL_SCALING>0</PARALLEL_SCALING>
        <OVERALL_PERFORMANCE>Normal</OVERALL_PERFORMANCE>
        <HOSTS_TO_SCAN>
          <EXTERNAL_SCANNERS>10</EXTERNAL_SCANNERS>
          <SCANNER_APPLIANCES>30</SCANNER_APPLIANCES>
        </HOSTS_TO_SCAN>
        <APPLICATIONS_TO_RUN>
```

<TOTAL PROCESSES>10</TOTAL PROCESSES>
<HTTP PROCESSES>10</HTTP PROCESSES>
</PROCESSES_TO_RUN>
<PACKET_DELAY>Medium</PACKET_DELAY>

<PORT SCANNING AND HOST_DISCOVERY>Normal</PORT SCANNING AND HOST_DISCOVERY>
</PERFORMANCE>
<DISSOLVABLE_AGENT>
<DISSOLVABLE_AGENT_ENABLE>0</DISSOLVABLE_AGENT_ENABLE>
</DISSOLVABLE_AGENT>

<HAS PASSWORD_AUDITING_ENABLE>0</HAS PASSWORD_AUDITING_ENABLE>
</PASSWORD_AUDITING_ENABLE>

<WINDOWS_SHARE_ENUMERATION_ENABLE>0</WINDOWS_SHARE_ENUMERATION_ENABLE>
</WINDOWS_SHARE_ENUMERATION_ENABLE>

<WINDOWS_DIRECTORY_SEARCH_ENABLE>0</WINDOWS_DIRECTORY_SEARCH_ENABLE>
</WINDOWS_DIRECTORY_SEARCH_ENABLE>

</DISSOLVABLE_AGENT>

<FILE_INTEGRITY_MONITORING>
<AUTO_UPDATE_EXPECTED_VALUE>1</AUTO_UPDATE_EXPECTED_VALUE>
</FILE_INTEGRITY_MONITORING>

<CONTROL_TYPES>
<FIM_CONTROLS_ENABLED>0</FIM_CONTROLS_ENABLED>
</CONTROL_TYPES>

</SCAN>

<ADDITIONAL>

<HOST_DISCOVERY>
<TCP_PORTS>
<STANDARD_SCAN>1</STANDARD_SCAN>
</TCP_PORTS>
<UDP_PORTS>
<STANDARD_SCAN>1</STANDARD_SCAN>
</UDP_PORTS>
</ICMP>1</ICMP>
</HOST_DISCOVERY>

<PACKET_OPTIONS>

<IGNORE_FIREWALL_GENERATED_TCP_RST>0</IGNORE_FIREWALL_GENERATED_TCP_RST>

<IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>0</IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>
<NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>0</NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>
</PACKET_OPTIONS>
</ADDITIONAL>
</OPTION_PROFILE>

<BASIC_INFO>
  <ID>31118</ID>
  <GROUP_NAME><![CDATA[pc 55]]></GROUP_NAME>
  <GROUP_TYPE>compliance</GROUP_TYPE>
  <USER_ID>![CDATA[John Smith (jsmith_ap)]]</USER_ID>
  <UNIT_ID>0</UNIT_ID>
  <SUBSCRIPTION_ID>10421401</SUBSCRIPTION_ID>
  <IS_GLOBAL>0</IS_GLOBAL>
  <UPDATE_DATE>2018-04-10T11:10:36Z</UPDATE_DATE>
</BASIC_INFO>

<SCAN>
  <PORTS>
    <TARGETED_SCAN>1</TARGETED_SCAN>
  </PORTS>
  <PERFORMANCE>
    <PARALLEL_SCALING>0</PARALLEL_SCALING>
    <OVERALL_PERFORMANCE>High</OVERALL_PERFORMANCE>
    <HOSTS_TO_SCAN>
      <EXTERNAL_SCANNERS>20</EXTERNAL_SCANNERS>
      <SCANNER_APPLIANCES>40</SCANNER_APPLIANCES>
    </HOSTS_TO_SCAN>
    <PROCESSES_TO_RUN>
      <TOTAL_PROCESSES>15</TOTAL_PROCESSES>
      <HTTP_PROCESSES>15</HTTP_PROCESSES>
    </PROCESSES_TO_RUN>
    <PACKET_DELAY>Short</PACKET_DELAY>
  </PERFORMANCE>
  <SCAN_RESTRICTION>
    <SCAN_BY_POLICY>
      <POLICY>
        <ID>10472</ID>
        <TITLE><![CDATA[jp]]></TITLE>
      </POLICY>
    </SCAN_BY_POLICY>
  </SCAN_RESTRICTION>
  <FILE_INTEGRITY_MONITORING>
    <AUTO_UPDATEEXPECTED_VALUE>1</AUTO_UPDATEEXPECTED_VALUE>
  </FILE_INTEGRITY_MONITORING>
</SCAN>
<ADDITIONAL>
  <HOST_DISCOVERY>
    <TCP_PORTS>
      <STANDARD_SCAN>1</STANDARD_SCAN>
      <TCP_ADDITIONAL>
        <HAS_ADDITIONAL>1</HAS_ADDITIONAL>
        <ADDITIONAL_PORTS>80,35</ADDITIONAL_PORTS>
      </TCP_ADDITIONAL>
    </TCP_PORTS>
    <UDP_PORTS>
      <STANDARD_SCAN>1</STANDARD_SCAN>
    </UDP_PORTS>
    <ICMP>1</ICMP>
  </HOST_DISCOVERY>
</ADDITIONAL>

<WATCHGUARD_DEFAULT_BLOCKED_PORTS>1</WATCHGUARD_DEFAULT_BLOCKED_PORTS>
  <ALL_REGISTERED_IPS>1</ALL_REGISTERED_IPS>
</BLOCK_RESOURCES>

<PACKET_OPTIONS>
  <IGNORE_FIREWALL_GENERATED_TCP_RST>1</IGNORE_FIREWALL_GENERATED_TCP_RST>
  <IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>1</IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>
  <NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>1</NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>
</PACKET_OPTIONS>

</OPTION_PROFILE>

<OPTION_PROFILE>
  <BASIC_INFO>
    <ID>51481401</ID>
    <GROUP_NAME><![CDATA[pc op - 1]]></GROUP_NAME>
    <GROUP_TYPE>compliance</GROUP_TYPE>
    <USER_ID><![CDATA[John Smith (jsmith_ap)]></USER_ID>
    <UNIT_ID>0</UNIT_ID>
    <SUBSCRIPTION_ID>10421401</SUBSCRIPTION_ID>
    <IS_GLOBAL>0</IS_GLOBAL>
    <UPDATE_DATE>2018-04-10T11:10:36Z</UPDATE_DATE>
  </BASIC_INFO>
</OPTION_PROFILE>

<SCAN>
  <PORTS>
<TARGETED_SCAN>1</TARGETED_SCAN>
</PORTS>

<PERFORMANCE>
<PARALLEL_SCALING>1</PARALLEL_SCALING>
<OVERALL_PERFORMANCE>High</OVERALL_PERFORMANCE>

<HOSTS_TO_SCAN>
<EXTERNAL_SCANNERS>20</EXTERNAL_SCANNERS>
<SCANNER_APPLIANCES>40</SCANNER_APPLIANCES>
</HOSTS_TO_SCAN>

<PROCESSES_TO_RUN>
<TOTAL_PROCESSES>15</TOTAL_PROCESSES>
<HTTP_PROCESSES>15</HTTP_PROCESSES>
</PROCESSES_TO_RUN>

<PACKET_DELAY>Short</PACKET_DELAY>

<PORT_SCANNING_AND_HOST_DISCOVERY>Normal</PORT_SCANNING_AND_HOST_DISCOVERY>
</PERFORMANCE>

<SCAN_RESTRICTION>
<SCAN_BY_POLICY>
<POLICY>
<ID>14487</ID>
<TITLE><![CDATA[jp2]]></TITLE>
</POLICY>
</SCAN_BY_POLICY>
</SCAN_RESTRICTION>

<FILE_INTEGRITY_MONITORING>
<AUTO_UPDATE_EXPECTED_VALUE>0</AUTO_UPDATE_EXPECTED_VALUE>
</FILE_INTEGRITY_MONITORING>
</SCAN>

<ADDITIONAL>

<HOST_DISCOVERY>

<TCP_PORTS>
<STANDARD_SCAN>1</STANDARD_SCAN>
<TCP_ADDITIONAL>
<HAS_ADDITIONAL>1</HAS_ADDITIONAL>
<ADDITIONAL_PORTS>1</ADDITIONAL_PORTS>
</TCP_ADDITIONAL>
</TCP_PORTS>

<UDP_PORTS>
<STANDARD_SCAN>1</STANDARD_SCAN>
</UDP_PORTS>

<ICMP>1</ICMP>
</HOST_DISCOVERY>

<BLOCK_RESOURCES>
<WATCHGUARD_DEFAULT_BLOCKED_PORTS>1</WATCHGUARD_DEFAULT_BLOCKED_PORTS>
   <ALL_REGISTERED_IPS>1</ALL_REGISTERED_IPS>
</BLOCK_RESOURCES>
<PACKET_OPTIONS>
<IGNORE_FIREWALL_GENERATED_TCP_RST>1</IGNORE_FIREWALL_GENERATED_TCP_RST>
<IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>1</IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>
<NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>1</NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>
</PACKET_OPTIONS>
</ADDITIONAL>
</OPTION_PROFILE>
</OPTION_PROFILE>

<SCAN>
   <PORTS>
      <STANDARD_SCAN>1</STANDARD_SCAN>
   </PORTS>

   <PERFORMANCE>
      <PARALLEL_SCALING>0</PARALLEL_SCALING>
      <OVERALL_PERFORMANCE>Normal</OVERALL_PERFORMANCE>
      <HOSTS_TO_SCAN>
         <EXTERNAL_SCANNERS>10</EXTERNAL_SCANNERS>
         <SCANNER_APPLIANCES>30</SCANNER_APPLIANCES>
      </HOSTS_TO_SCAN>
      <HOSTS_TO_SCAN>
      </HOSTS_TO_SCAN>
   </PERFORMANCE>

   <HOSTS_TO_SCAN>
      <TOTAL_PROCESSES>10</TOTAL_PROCESSES>
      <HTTP_PROCESSES>10</HTTP_PROCESSES>
   </HOSTS_TO_SCAN>
   <PROCICES_TO_RUN>
      <TOTAL_PROCESSES>10</TOTAL_PROCESSES>
      <HTTP_PROCESSES>10</HTTP_PROCESSES>
   </PROCICES_TO_RUN>
   <PACKET_DELAY>Medium</PACKET_DELAY>

   <PORT_SCANNING_AND_HOST_DISCOVERY>Normal</PORT_SCANNING_AND_HOST_DISCOVERY>
</SCAN>
<OPTION_PROFILE>
   <ID>14661401</ID>
   <TITLE><![CDATA[policy - 2]]></TITLE>
</OPTION_PROFILE>

<OPTION_PROFILE>
   <ID>14651401</ID>
   <TITLE><![CDATA[policy - 1]]></TITLE>
</OPTION_PROFILE>

<FILE_INTEGRITY_MONITORING>
   <AUTO_UPDATE_EXPECTED_VALUE>0</AUTO_UPDATE_EXPECTED_VALUE>
</FILE_INTEGRITY_MONITORING>

<ADDITIONAL>
   <HOST_DISCOVERY>
      <TCP_PORTS>
         <STANDARD_SCAN>1</STANDARD_SCAN>
      </TCP_PORTS>
      <UDP_PORTS>
         <CUSTOM_PORT><![CDATA[37,53,68,69,111]]></CUSTOM_PORT>
      </UDP_PORTS>
      <ICMP>1</ICMP>
   </HOST_DISCOVERY>
   <BLOCK_RESOURCES>
      <CUSTOM_PORT_LIST><![CDATA[111]]></CUSTOM_PORT_LIST>
      <CUSTOM_IP_LIST><![CDATA[10.10.10.6]]></CUSTOM_IP_LIST>
   </BLOCK_RESOURCES>
   <PACKET_OPTIONS>
      <IGNORE_FIREWALL_GENERATED_TCP_RST>0</IGNORE_FIREWALL_GENERATED_TCP_RST>
      <IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>0</IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK>
      <NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>0</NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY>
   </PACKET_OPTIONS>
</ADDITIONAL>
</OPTION_PROFILES>
Sample - List Option Profile for Database UDCs

List the database preference key setting and it's corresponding value in Option Profile for MS SQL, Oracle, Sybase, PostgreSQL/Pivotal Greenplum, SAP IQ, and IBM DB2.

API request:
```
curl -u "username:password" -H "X-Requested-With:curl" -H "Content-type: text/xml" -X -d "action=list&id=1710150" "https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pc/"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE OPTION_PROFILES SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/info.dtd">
<OPTION_PROFILES>
  <OPTION_PROFILE>
    <BASIC_INFO>
      <ID>1710150</ID>
      ...
    </SCAN_BY_POLICY>
    </SCAN_RESTRICTION>
    <DATABASE_PREFERENCE_KEY>
      <MSSQL>
        <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
        <DB_UDC_LIMIT>250</DB_UDC_LIMIT>
      </MSSQL>
      <ORACLE>
        <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
        <DB_UDC_LIMIT>10</DB_UDC_LIMIT>
      </ORACLE>
      <SYBASE>
        <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
        <DB_UDC_LIMIT>60</DB_UDC_LIMIT>
      </SYBASE>
      <POSTGRESQL>
        <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
        <DB_UDC_LIMIT>2500</DB_UDC_LIMIT>
      </POSTGRESQL>
      <DB2>
        <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
        <DB_UDC_LIMIT>300</DB_UDC_LIMIT>
      </DB2>
    </DATABASE_PREFERENCE_KEY>
  </OPTION_PROFILE>
</OPTION_PROFILES>
```
Scan Configuration
Option Profiles for Compliance

<AUTO_UPDATEEXPECTED_VALUE>0</AUTO_UPDATEEXPECTED_VALUE>
</FILE_INTEGRITY_MONITORING>
</SCAN>
...
</OPTION_PROFILES>

DTD
<platform API server>/api/2.0/fo/subscription/option_profile/option_profile_info.dtd

Delete Compliance Option Profile
/ap/2.0/fo/subscription/option_profile/pc/?action=delete
[GET] [POST]

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) The ID of the option profile.</td>
</tr>
</tbody>
</table>

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST "action=delete&id=51491401" "http://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pc/"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "http://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-10T11:10:36Z</DATETIME>
    <TEXT>Option Profile Deleted Successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>51491401</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
Download a list of vulnerabilities from Qualys’ KnowledgeBase. Several input parameters grant users control over which vulnerabilities to download and the amount of detail to download, and the XML output provides a rich information source for each vulnerability.

Qualys’ Sofware-as-a-Service (SaaS) technology includes its KnowledgeBase, with the industry’s largest number of vulnerability signatures, that is continuously updated by Qualys’ Research and Development team. Qualys is fully dedicated to providing the most accurate security audits in the industry. Each day new and updated signatures are tested in Qualys’ own vulnerability labs and then published, making them available to Qualys customers. When Threat Protection is enabled for your subscription, the output will include Real-Time Threat Indicators (RTIs) associated with vulnerabilities.

Authorized Qualys users have the ability to download vulnerability data using the KnowledgeBase API. Please contact Qualys Support or your sales representative if you would like to obtain authorization for your subscription.

Permissions - Your subscription must be granted permission to run this API function. Please contact Qualys Support or your sales representative to receive this authorization.

### Input Parameters

Several optional input parameters may be specified. When unspecified, the XML output includes all vulnerabilities in the KnowledgeBase, showing basic details for each vulnerability. Several optional parameters allow you specify filters. When filter parameters are specified, these parameters are ANDed by the service to filter the data from the output.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required) Show (echo) the request’s input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Optional) Used to filter the XML output to include only vulnerabilities that have QID numbers matching the QID numbers you specify.</td>
</tr>
<tr>
<td>Note:</td>
<td>If the QID numbers and number range specified are too large, it is recommended to use id_min={value} and id_max={value} parameters.</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities that have a QID number greater than or equal to a QID number you specify.</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities that have a QID number less than or equal to a QID number you specify.</td>
</tr>
<tr>
<td>is_patchable={0</td>
<td>1}</td>
</tr>
<tr>
<td>last_modified_after={date}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities last modified after a certain date and time. When specified vulnerabilities last modified by a user or by the service will be shown. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).</td>
</tr>
<tr>
<td>last_modified_before={date}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities last modified before a certain date and time. When specified vulnerabilities last modified by a user or by the service will be shown. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).</td>
</tr>
<tr>
<td>last_modified_by_user_after={date}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities last modified by a user after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).</td>
</tr>
<tr>
<td>last_modified_by_user_before={date}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities last modified by a user before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).</td>
</tr>
<tr>
<td>last_modified_by_service_after={date}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities last modified by the service after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>last_modified_by_service_before</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities last modified by the service before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).</td>
</tr>
<tr>
<td>published_after={date}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities published after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).</td>
</tr>
<tr>
<td>published_before={date}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities published before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).</td>
</tr>
<tr>
<td>discovery_method={value}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities assigned a certain discovery method. A valid value is: Remote, Authenticated, RemoteOnly, AuthenticatedOnly, or RemoteAndAuthenticated. When “Authenticated” is specified, the service shows vulnerabilities that have at least one associated authentication type. Vulnerabilities that have at least one authentication type can be detected in two ways: 1) remotely without using authentication, and 2) using authentication.</td>
</tr>
<tr>
<td>discovery_auth_types={value}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities having one or more authentication types. A valid value is: Windows, Oracle, Unix, SNMP, DB2, HTTP, PANOS, TOMCAT, MARIADB, MongoDB, WEBLOGIC, MySQL, VMware. Multiple values should be comma-separated.</td>
</tr>
<tr>
<td>show_pci_reasons={0</td>
<td>1}</td>
</tr>
<tr>
<td>show_supported_modules_info={0</td>
<td>1}</td>
</tr>
<tr>
<td>show_disabled_flag={0</td>
<td>1}</td>
</tr>
<tr>
<td>show_qid_change_log={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
Real-Time Threat Indicators (RTIs)

The KnowledgeBase list output includes Real-Time Threat Indicators (RTIs) associated with each vulnerability. RTIs appear as part of vulnerability details under THREAT_INTELLIGENCE. Please note that RTIs are only visible when Threat Protection is enabled for the subscription.

Real-Time Threat Indicators are described below.

<table>
<thead>
<tr>
<th>RTI (ID)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero_Day (1)</td>
<td>Active attack has been observed in the wild and there is no patch from the vendor. An active attack is a prerequisite for this RTI in addition to no patch from the vendor. If a vulnerability is not actively attacked this RTI will not be set (even if there is no patch from the vendor). If a patch becomes available Qualys will remove the Zero Day RTI attribute which helps users to focus only on vulnerabilities that are actively exploited and there is no official patch.</td>
</tr>
<tr>
<td>Exploit_Public (2)</td>
<td>Exploit knowledge is well known and a working exploitation code is publicly available. Potential of active attacks is very high. This attribute is set for example when PoC exploit code is available from Exploit-DB, Metasploit, Core, Immunity or other exploit vendors. This RTI does not necessarily indicate that active attacks have been observed in the wild.</td>
</tr>
<tr>
<td>Active_Attacks (3)</td>
<td>Active attacks have been observed in the wild. This information is derived from Malware, Exploit Kits, acknowledgment from vendors, US-CERT and similar trusted sources. If there are no patches, Qualys will mark it as Zero Day, in addition, to actively attacked.</td>
</tr>
<tr>
<td>High_Lateral_Movement (4)</td>
<td>After a successful compromise, the attacker has high potential to compromise other machines in the network.</td>
</tr>
<tr>
<td>Easy_Exploit (5)</td>
<td>The attack can be carried out easily and requires little skills or does not require additional information.</td>
</tr>
<tr>
<td>High_Data_Loss (6)</td>
<td>Successful exploitation will result in massive data loss on the host.</td>
</tr>
<tr>
<td>Denial_of_Service (7)</td>
<td>Successful exploitation will result in denial of service.</td>
</tr>
<tr>
<td>No_Patch (8)</td>
<td>The vendor has not provided an official fix.</td>
</tr>
<tr>
<td>Malware (9)</td>
<td>Malware has been associated with the vulnerability.</td>
</tr>
<tr>
<td>Exploit_Kit (10)</td>
<td>Exploit Kit has been associated with this vulnerability. Exploit Kits are usually cloud based toolkits that help malware writers in identifying vulnerable browsers/plugins and install malware. Users can also search on Exploit Kit name like Angler, Nuclear, Rig and others.</td>
</tr>
<tr>
<td>Wormable (11)</td>
<td>Wormable has been associated with this vulnerability. The vulnerability can be used in “worms” - malware that spreads itself without user interaction.</td>
</tr>
</tbody>
</table>
Predicted_High_Risk (12) | Predicted High Risk has been associated with this vulnerability. Qualys Machine Learning Model predicted this vulnerability as a High Risk based on various data sources including NVD, Social network, Dark web, Security Blogs, Code repository, Exploits, etc.

Privilege_Escalation (13) | Successful exploitation allows an attacker to gain elevated privileges.

Unauthenticated_Explotation (14) | Exploitation of this vulnerability does not require authentication.

Remote_Code_Execution (15) | Successful exploitation allows an attacker to execute arbitrary commands or code on a targeted system or in a target process.

Ransomware (16) | This vulnerability has been exploited in attack vectors where ransomware has been deployed. In other words, this vulnerability is associated with known ransomware.

Solorigate_Sunburst (17) | Solorigate Sunburst has been associated with all the CVEs, used by FireEye’s Red Team tools to test the security of their client environments and compromised versions of SolarWinds Orion.

CISA Known Exploited Vulnerabilities (18) | CISA maintains a catalog of the top publicly known vulnerabilities being exploited in the wild and organizations (referred as CISA Known Exploitable Vulnerabilities) are advised to patch affected systems on priority. This RTI indicates that the vulnerability is associated with the CISA catalog and with CVE mappings to respective QIDs. We will add the CISA Known Exploited Vulnerabilities to QIDs within 24hrs of CISA catalog updates with new CVEs.

This [CISA Directive](#) recommends urgent and prioritized remediation of the vulnerabilities that adversaries are actively exploiting.

The timelines are available in [CISA’s Catalog](#) for each of the CVEs.

### Samples

Sample 1 - Request all vulnerabilities in the KnowledgeBase showing basic details:

```bash
curl -u "user:password" -H "X-Requested-With: Curl" -X "POST" -d "action=list" "https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" > output.txt
```

Sample 2 - Request patchable vulnerabilities that have QIDs 1-200 showing all details:

```bash
curl -u "user:password" -H "X-Requested-With: Curl" -X "POST" -d "action=list&ids=1-200&is_patchable=1&details=All" "https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" > output.txt
```
Sample 3 - Request vulnerabilities that were last modified by the service after July 20, 2011 and that have the “remote and authenticated” discovery method:

```
curl -u "user:password" -H "X-Requested-With: Curl" -X "POST"
-d "action=list&last_modified_by_service_after=2011-07-20
&discovery_method=RemoteAndAuthenticated"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" >
output.txt
```

**DTD**

```
<platform API
server>/api/2.0/fo/knowledge_base/vuln/knowledge_base_vuln_list_output.dtd
```
KnowledgeBase QVS Download in JSON Format

/\api/2.0/fo/knowledge_base/qvs/?action=list

[GET] [POST]

Use this API endpoint to download Qualys Vulnerability Score (QVS) information from the Vulnerability KnowledgeBase for one or more CVE IDs based on certain criteria. The API output is in JSON format.

The Qualys Vulnerability Score (QVS) is a Qualys-assigned score to a vulnerability based on multiple factors associated with the CVE, such as CVSS and external threat indicators like active exploitation, exploit code maturity and much more.

Input Parameters

Use the following input parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required) You must specify the list action.</td>
</tr>
<tr>
<td>details=[Basic</td>
<td>All]</td>
</tr>
<tr>
<td>cve={value}</td>
<td>(Required) Filter the JSON output to only show vulnerabilities associated with the CVE IDs that you specify.</td>
</tr>
<tr>
<td>qvs_last_modified _before={date}</td>
<td>(Optional) Show only CVE IDs with a QVS score that was last modified before a certain date and time. Valid date format is: YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2021-12-01” or “2021-1201T23:12:00Z”.</td>
</tr>
<tr>
<td>qvs_last_modified _after={date}</td>
<td>(Optional) Show only CVE IDs with a QVS score that was last modified after a certain date and time. Valid date format is: YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2021-12-01” or “2021-12-01T23:12:00Z”.</td>
</tr>
<tr>
<td>qvs_min={value}</td>
<td>(Optional) Show only CVEs with a QVS value greater than or equal to the QVS min value specified. (QVS Prime will not be considered.) When qvs_min and qvs_max are specified in the same request, the qvs_min value must be less than the qvs_max value.</td>
</tr>
<tr>
<td>qvs_max={value}</td>
<td>(Optional) Show only CVEs with a QVS value less than or equal to the QVS max value specified. (QVS Prime will not be considered.) When qvs_min and qvs_max are specified in the same request, the qvs_min value must be less than the qvs_max value.</td>
</tr>
<tr>
<td>nvd_published_before={date}</td>
<td>(Optional) Show only CVE IDs with a QVS score that was published before a certain date and time. Valid date format is: YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2021-12-01” or “2021-1201T23:12:00Z”.</td>
</tr>
<tr>
<td>nvd_published_after={date}</td>
<td>(Optional) Show only CVE IDs with a QVS score that was published after a certain date and time. Valid date format is: YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2021-12-01” or “2021-12-01T23:12:00Z”.</td>
</tr>
</tbody>
</table>
Sample KnowledgeBase QVS Download

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/qvs/?action=list&
cve=CVE-2021-36765,CVE-2021-36798&qvs_min=1&qvs_max=100&qvs_last_modified_after=2016-12-16T05:00:17Z&qvs_last_modified_before=2022-01-20T05:00:17Z&nvd_published_after=2016-12-16T05:00:17Z&nvd_published_before=2022-12-16T05:00:17Z&details=All"

JSON output:

```json
{
  "CVE-2021-36765": {
    "base": {
      "id": "CVE-2021-36765",
      "idType": "CVE",
      "qvs": "28",
      "qvsLastChangedDate": 1642032000,
      "nvdPublishedDate": 1628086500
    },
    "contributingFactors": {
      "cvss": "5",
      "cvssVersion": "v2"
    }
  },
  "CVE-2021-36798": {
    "base": {
      "id": "CVE-2021-36798",
      "idType": "CVE",
      "qvs": "78",
      "qvsLastChangedDate": 1642550400,
      "nvdPublishedDate": 1628514900
    },
    "contributingFactors": {
      "cvss": "5",
      "cvssVersion": "v2",
      "exploitMaturity": ["poc"
    ]
  }
}
```
Editing Vulnerabilities

/api/2.0/fo/knowledge_base/vuln/

[POST]

Edit, reset and list the edited vulnerabilities in the Qualys Vulnerability KnowledgeBase.

Permissions - Managers have permissions to edit vulnerabilities and make API requests to edit a vulnerability, reset a vulnerability and list customized vulnerabilities.

Edit a vulnerability

You can change the severity level and/or add comments to Threat, Impact or Solution. Providing at least one optional parameter is mandatory.

### Parameter | Description
--- | ---
action=edit | (Required) POST method is required
qid={value} | (Required) QID of the vulnerability to be edited.
severity={value} | (Optional) Severity level between 1 to 5. Changing the severity level of a vulnerability impacts how the vulnerability appears in reports and how it is eventually prioritized for remediation. For example, by changing a vulnerability from a severity 2 to a severity 5, remediation tickets for the vulnerability could have a higher priority and shorter deadline for resolution.
disable={0|1} | (Optional) Specify 1 to disable the vulnerability. Default is 0. When you disable a vulnerability it is globally filtered out from all hosts in all scan reports. The vulnerability is also filtered from host information, asset search results and your dashboard. You may include disabled vulnerabilities in scan reports by changing report filter settings.
threat_comment | (Optional) Threat comments in plain text.
impact_comment | (Optional) Impact comments in plain text.
solution_comment | (Optional) Solution comments in plain text.

Comments added for Threat, Impact, or Solution are appended to the service-provided descriptions in the vulnerability details.

**API request:**

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"action=edit&impact_comment=testimpact&qid=27014"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
```
Reset a vulnerability
You can change the vulnerability settings back to original.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=set</td>
<td>(Required) POST method is required</td>
</tr>
<tr>
<td>qid={value}</td>
<td>(Required) QID of the vulnerability to be reset.</td>
</tr>
</tbody>
</table>

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" action=set&qid=27014
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2017-03-02T08:55:11Z</DATETIME>
    <TEXT>Custom Vuln Data has been reset successfully</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

List customized vulnerabilities
You can list the vulnerabilities that are edited.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=custom</td>
<td>(Required) GET or POST method can be used.</td>
</tr>
</tbody>
</table>
API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"action=custom"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE KB_CUSTOM_VULN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/kb_custom_vuln_list_output.dtd">
<KB_CUSTOM_VULN_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-03-02T08:47:52Z</DATETIME>
    <CUSTOM_VULN_LIST>
      <CUSTOM_VULN_DATA>
        <QID>
          <![CDATA[27014]]>
        </QID>
        <SEVERITY_LEVEL>5</SEVERITY_LEVEL>
        <ORIGINAL_SEVERITY_LEVEL>5</ORIGINAL_SEVERITY_LEVEL>
        <IS_DISABLED>1</IS_DISABLED>
        <UPDATED_DATETIME>
          <![CDATA[2017-03-02T05:40:02Z]]>
        </UPDATED_DATETIME>
        <UPDATED_BY>
          <![CDATA[mr_md]]>
        </UPDATED_BY>
        <THREAT_COMMENT>
          <![CDATA[threat123]]>
        </THREAT_COMMENT>
        <IMPACT_COMMENT>
          <![CDATA[impact123]]>
        </IMPACT_COMMENT>
        <SOLUTION_COMMENT>
          <![CDATA[solution123]]>
        </SOLUTION_COMMENT>
      </CUSTOM_VULN_DATA>
    </CUSTOM_VULN_LIST>
  </RESPONSE>
</KB_CUSTOM_VULN_LIST_OUTPUT>
```

DTD
```
<platform API server>/api/2.0/fo/knowledge_base/vuln/kb_custom_vuln_list_output.dtd
```
**Static Search Lists**

`/api/2.0/fo/qid/search_list/static/`

Create static search lists and get information about them.

Permissions - as below.

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager, Unit Manager, Scanner, Reader</td>
<td>Create, update, list and delete search lists.</td>
</tr>
<tr>
<td>Auditor</td>
<td>No permission to create, update, list and delete search lists.</td>
</tr>
</tbody>
</table>

**List static search lists**

**Input parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={id1,id2...}</td>
<td>(Optional) One or more search list IDs to display. Multiple IDs are comma separated.</td>
</tr>
</tbody>
</table>

**Sample - List static search list**

**API request:**

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" "https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/static/?action=list&ids=381"
```

**XML response:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE STATIC_SEARCH_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/static/static_list_output.dtd">  
<STATIC_SEARCH_LIST_OUTPUT>  
  <RESPONSE>  
    <DATETIME>2018-06-06T06:20:03Z</DATETIME>  
    <STATIC_LISTS>  
      <STATIC_LIST>  
        <ID>381</ID>  
        <TITLE><![CDATA[static search list]]></TITLE>  
        <GLOBAL>Yes</GLOBAL>  
        <OWNER>acme_tb</OWNER>  
        <CREATED><![CDATA[06/01/2018 at 15:18:42 (GMT+0530)]]></CREATED>  
      </STATIC_LIST>  
    </STATIC_LISTS>  
  </RESPONSE>  
</STATIC_SEARCH_LIST_OUTPUT>  
```
<MODIFIED_BY>acme_tb</MODIFIED_BY>
<MODIFIED>![CDATA[06/02/2018 at 15:18:42 (GMT+0530)]]</MODIFIED>

<QIDS>
  <QID>1000</QID>
  <QID>1001</QID>
</QIDS>

<!-- This list is used in the following option profiles //-->

<OPTION_PROFILES>
  <OPTION_PROFILE>
    <ID>135</ID>
    <TITLE>![CDATA[Initial Options]]</TITLE>
  </OPTION_PROFILE>
</OPTION_PROFILES>

<!-- This list is used in the following report templates //-->

<REPORT_TEMPLATES>
  <REPORT_TEMPLATE>
    <ID>256</ID>
    <TITLE>![CDATA[Scan Report Template]]</TITLE>
  </REPORT_TEMPLATE>
</REPORT_TEMPLATES>

<!-- This list is used in the following remediation policies. //-->

<REMEDIATION_POLICIES>
  <REMEDIATION_POLICY>
    <ID>655</ID>
    <TITLE>![CDATA[Remediation Policy 1]]</TITLE>
  </REMEDIATION_POLICY>
</REMEDIATION_POLICIES>

<!-- This search list is associated with following distribution groups. //-->

<DISTRIBUTION_GROUPS>
  <DISTRIBUTION_GROUP>
    <NAME>![CDATA[All]]</NAME>
  </DISTRIBUTION_GROUP>
</DISTRIBUTION_GROUPS>

<COMMENTS>![CDATA[This is my first comment for this list]]</COMMENTS>
</STATIC_LIST>
</STATIC_LISTS>
</RESPONSE>
</SEARCH_LIST_OUTPUT>
DTD

<platform API server>/api/2.0/fo/qid/search_list/static/static_list_output.dtd

Create static search lists

Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=create</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0/1</td>
<td>(Optional) Specify 1 to show input parameters in XML output.</td>
</tr>
<tr>
<td>title=value</td>
<td>(Required) A user defined search list title. Maximum is 256 characters (ascii).</td>
</tr>
<tr>
<td>qids=num1,num2...</td>
<td>(Required) QIDs to include in the search list. Ranges are allowed.</td>
</tr>
<tr>
<td>global=0/1</td>
<td>(Optional) Specify 1 to make this a global search list, available to all subscription users.</td>
</tr>
<tr>
<td>comments=value</td>
<td>(Optional) User defined comments.</td>
</tr>
</tbody>
</table>

Sample - Create search list

API request:

curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=create&title=My+Static+Search+List&qids=68518-68522,48000" "https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/static/"

XML response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2015-09-01T21:32:40Z</DATETIME>
    <TEXT>New search list created successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>136992</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
# Update static search list

## Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>action=update</code></td>
<td>(Required)</td>
</tr>
<tr>
<td>`echo_request=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>id=[id]</code></td>
<td>(Required) The ID of the search list you want to update.</td>
</tr>
<tr>
<td><code>title=[value]</code></td>
<td>(Optional) The search list title. Maximum is 256 characters (ascii).</td>
</tr>
<tr>
<td>`global=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>qids=(num1, num2...)</code></td>
<td>(Optional) QIDs/ranges to include in the search list. Multiple entries are comma separated. &quot;&quot;&quot;QIDs specified will replace all existing ones defined for the search list, if any. QIDs cannot be specified with add_qids or remove_qids in the same request.</td>
</tr>
<tr>
<td><code>add_qids=(num1, num2...)</code></td>
<td>(Optional) QIDs/ranges you want to add to the existing ones defined for the search list. When the same QIDs are passed using add_qids and remove_qids in the same request, the QIDs are added to the list. Add_qids cannot be specified with qids in the same request.</td>
</tr>
<tr>
<td><code>remove_qids=(num1, num2...)</code></td>
<td>(Optional) QIDs/ranges you want to remove the existing ones defined for the search list. When the same QIDs are passed using add_qids and remove_qids in the same request, the QIDs are added to the list. Remove_qids cannot be specified with qids in the same request.</td>
</tr>
<tr>
<td><code>comments=[value]</code></td>
<td>(Optional) User defined comments.</td>
</tr>
</tbody>
</table>

## Sample - Update static search list

**API request:**
```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=update&id=136992&global=1&qids=68518-68522,48000-48004" "https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/static/"
```

**XML response:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2015-09-01T21:32:40Z</DATETIME>
    <TEXT>Search list updated successfully</TEXT>
    <ITEM_LIST>
```

231
Delete static search list

Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>id={id}</td>
<td>(Required) The ID of the search list you want to delete.</td>
</tr>
</tbody>
</table>

Sample - Delete static search list

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=delete&id=136992"
"https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/static/"
```

XML response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2015-09-01T21:32:40Z</DATETIME>
    <TEXT>search list deleted successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>136992</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Dynamic Search Lists

/api/2.0/fo/qid/search_list/dynamic/

Create dynamic search lists and get information about them.

Permissions - as described below

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager, Unit Manager, Scanner, Reader</td>
<td>Create, update, list and delete search lists.</td>
</tr>
<tr>
<td>Auditor</td>
<td>No permission to create, update, list and delete search lists.</td>
</tr>
</tbody>
</table>

List dynamic search lists

Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={id1,id2...}</td>
<td>(Optional) One or more search list IDs to display. Multiple IDs are comma separated.</td>
</tr>
<tr>
<td>show_qids= {0</td>
<td>1}</td>
</tr>
<tr>
<td>show_option_profiles= {0</td>
<td>1}</td>
</tr>
<tr>
<td>show_distribution_groups= {0</td>
<td>1}</td>
</tr>
<tr>
<td>show_report_templates= {0</td>
<td>1}</td>
</tr>
<tr>
<td>show_remediation_policies= {0</td>
<td>1}</td>
</tr>
</tbody>
</table>

Sample - List dynamic search list

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" "https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/dynamic/?action=list&ids=381"

XML response:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE DYNAMIC_SEARCH_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/dynamic/dynamic_list_output.dtd">
<SEARCH_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2015-01-06T06:20:03Z</DATETIME>
    <DYNAMIC_LISTS>
      <DYNAMIC_LIST>
        <ID>381</ID>
        <TITLE><![CDATA[static search list]]></TITLE>
        <GLOBAL>Yes</GLOBAL>
        <OWNER>acme_tb</OWNER>
        <CREATED><![CDATA[07/27/2015 at 15:18:42 (GMT+0530)]]]</CREATED>
        <MODIFIED_BY>acme_tb</MODIFIED_BY>
        <MODIFIED><![CDATA[07/27/2015 at 15:18:42 (GMT+0530)]]]</MODIFIED>
        <QIDS>
          <QID>1000</QID>
          <QID>1001</QID>
        </QIDS>
        <CRITERIA>
          <VULNERABILITY_TITLE><![CDATA[NOT Title]]></VULNERABILITY_TITLE>
          <DISCOVERY_METHOD><![CDATA[Authenticated Only]]></DISCOVERY_METHOD>
          <AUTHENTICATION_TYPE><![CDATA[HTTP, Oracle, Unix]]></AUTHENTICATION_TYPE>
          <USER_CONFIGURATION><![CDATA[Disabled, Edited]]></USER_CONFIGURATION>
          <CATEGORY><![CDATA[NOT Backdoors and trojan horses, DNS and BIND]]></CATEGORY>
          <CONFIRMED_SEVERITY><![CDATA[1, 2]]></CONFIRMED_SEVERITY>
          <POTENTIAL_SEVERITY><![CDATA[2, 3]]></POTENTIAL_SEVERITY>
          <INFORMATION_SEVERITY><![CDATA[4, 5]]></INFORMATION_SEVERITY>
          <VENDOR><![CDATA[2brightsparks, 3com, 4d]]></VENDOR>
          <PRODUCT><![CDATA[.net_framework]]></PRODUCT>
          <CVSS_BASE_SCORE><![CDATA[2]]></CVSS_BASE_SCORE>
          <CVSS_TEMPORAL_SCORE><![CDATA[3]]></CVSS_TEMPORAL_SCORE>
          <CVSS_ACCESSVECTOR><![CDATA[Adjacent Network]]></CVSS_ACCESS_VECTOR>
          <PATCH_AVAILABLE><![CDATA[Yes, No]]></PATCH_AVAILABLE>
          <VIRTUAL_PATCH_AVAILABLE><![CDATA[Yes]]></VIRTUAL_PATCH_AVAILABLE>
      </DYNAMIC_LIST>
    </DYNAMIC_LISTS>
  </RESPONSE>
</SEARCH_LIST_OUTPUT>
<CVE_ID><![CDATA[NOT CVE]]></CVE_ID>
<CVE_ID_FILTER><![CDATA[Contains]]></CVE_ID_FILTER>
<EXPLOITABILITY><![CDATA[ExploitKits, Immunity - Dsquare]]></EXPLOITABILITY>
<ASSOCIATED_MALWARE><![CDATA[Trend Micro]]></ASSOCIATED_MALWARE>
<VENDOR_REFERENCE><![CDATA[NOT Linux]]></VENDOR_REFERENCE>
<BUGTRAQ_ID><![CDATA[NOT 15656]]></BUGTRAQ_ID>
<VULNERABILITY_DETAILS><![CDATA[details]]></VULNERABILITY_DETAILS>
<COMPLIANCE_DETAILS><![CDATA[details]]></COMPLIANCE_DETAILS>
<COMPLIANCE_TYPE><![CDATA[PCI, CobIT, HIPAA, GLBA, SOX]]></COMPLIANCE_TYPE>
<QUALYS_TOP_20><![CDATA[Top Internal 10, Top External 10]]></QUALYS_TOP_20>
<OTHER><![CDATA[Not exploitable due to configuration, Non-running services, 2008 SANS 20]]></OTHER>
<NETWORK_ACCESS><![CDATA[NAC / NAM]]></NETWORK_ACCESS>
<PUBLISHED><![CDATA[NOT 06/02/2015-07/20/2015]]></PUBLISHED>
<SERVICE_MODIFIED><![CDATA[NOT Previous 1 week]]></SERVICE_MODIFIED>
</CRITERIA>
</CRITERIA>
<!-- This list is used in the following option profiles //-->

<OPTION_PROFILES>
<OPTION_PROFILE>
<ID>135</ID>
<TITLE><![CDATA[Initial Options]]></TITLE>
</OPTION_PROFILE>
</OPTION_PROFILES>
<!-- This list is used in the following report templates //-->
<ID>655<ID>
<TITLE><![CDATA[Remediation Policy 1]]></TITLE>
</REMEDICATION_POLICY>
</REMEDICATION_POLICIES>
<!-- This search list is associated with following
distribution groups. //-->
<DISTRIBUTION_GROUPS>
<DISTRIBUTION_GROUP>
<ID>226<ID>
<TITLE><![CDATA[All]]></TITLE>
</DISTRIBUTION_GROUP>
</DISTRIBUTION_GROUPS>
</DYNAMIC_LIST>
</DYNAMIC_LISTS>
</RESPONSE>
</SEARCH_LIST_OUTPUT>

DTD
<platform API server>/api/2.0/fo/qid/search_list/dynamic/dynamic_list_output.dtd

Create dynamic search list

Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=create</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request</td>
<td>[0,1] (Optional) Specify 1 to show input parameters in XML output.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required) A user defined search list title. Maximum is 256 characters (ascii).</td>
</tr>
<tr>
<td>global=[0,1]</td>
<td>(Optional) Specify 1 to make this a global search list, available to all</td>
</tr>
<tr>
<td></td>
<td>subscription users.</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) User defined comments.</td>
</tr>
<tr>
<td>criteria</td>
<td>(Required) User defined search criteria. See “Search criteria”</td>
</tr>
</tbody>
</table>
Search criteria
Use these parameters to define search criteria for dynamic search lists, using create and update requests. All parameters act as vulnerability filters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>vuln_title=[value]</td>
<td>Vulnerability title (string); to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>not_vuln_title=[0</td>
<td>1]</td>
</tr>
<tr>
<td>discovery_methods=[value]</td>
<td>One or more discovery methods: Remote, Authenticated, Remote_Authenticated; by default all methods are included</td>
</tr>
<tr>
<td>auth_types=[value]</td>
<td>One or more of these authentication types: Windows, Unix, Oracle, SNMP, VMware, DB2, HTTP, MySQL, PANOS, TOMCAT, MARIADB, MongoDB, WEBLOGIC; multiple values are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>user_configuration=[value]</td>
<td>One or more of these user configuration values: disabled, custom; multiple values are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>categories=[value]</td>
<td>One or more vulnerability category names (strings); to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>not_categories=[0</td>
<td>1]</td>
</tr>
<tr>
<td>confirmed_severities=[value]</td>
<td>One or more confirmed vulnerability severities (1-5); multiple severities are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>potential_severities=[value]</td>
<td>One or more potential vulnerability severities (1-5); multiple severities are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>ig_severities=[value]</td>
<td>One or more information gathered severities (1-5); multiple severities are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>vendor_ids=[value]</td>
<td>One or more vendor IDs; multiple IDs are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>not_vendor_ids=[0</td>
<td>1]</td>
</tr>
<tr>
<td>products=[value]</td>
<td>Vendor product names; multiple names are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>not_products=[0</td>
<td>1]</td>
</tr>
<tr>
<td>patch_available={value}</td>
<td>Vulnerabilities with patches: 0 (no), 1 (yes); by default all vulnerabilities with and without patches are included; multiple values are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>virtual_patch_available={value}</td>
<td>Vulnerabilities with Trend Micro virtual patches: 0 (no), 1 (yes); by default vulnerabilities with and without these virtual patches are included; multiple values are comma separated; to unset value use update request and set to empty value</td>
</tr>
</tbody>
</table>
| cve_ids_filter                      | (Optional) Filter CVE IDs with the “Exact Match” or “Contains” search criteria:  
- Set to 1 to filter the CVE IDs that match exactly with the input CVE strings.  
- Set to 2 to filter the CVE IDs that contain the input CVE string. |
<p>| cve_ids={value}                     | One or more CVE IDs; multiple IDs are comma separated; to unset value use update request and set to empty value |
| not_cve_ids=[0|1]                   | Set to 1 for CVE IDs that do not match cve_ids parameter values                                 |
| exploitability={value}              | One or more vendors with exploitability info; multiple references are comma separated; to unset value use update request and set to empty value |
| malware_associated={value}          | One or more vendors with malware info; multiple references are comma separated; to unset value use update request and set to empty value |
| vendor_refs={value}                 | One or more vendor references; multiple vendors are comma separated; to unset value use update request and set to empty value |
| not_vendor_refs=[0|1]               | Set to 1 for vendor references that do not match vendor_refs parameter values                  |
| bugtraq_id={value}                  | Vulnerabilities with a Bugtraq ID number; to unset value use update request and set to empty value |
| not_bugtraq_id=[0|1]                | Set to 1 for vulnerabilities with Bugtraq IDs that do not match the bugtraq_id parameter value  |
| vuln_details={value}                | A string matching vulnerability details; to unset value use update request and set to empty value |
| compliance_details={value}          | A string matching compliance details; to unset value use update request and set to empty value |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>supported_modules={value}</td>
<td>One or more of these Qualys modules: VM, CA-Windows Agent, CA-Linux Agent, WAS, WAF, MD; multiple values are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>compliance_types={value}</td>
<td>One or more compliance types: PCI, CobiT, HIPAA, GLBA, SOX; multiple values are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>qualys_top_lists={value}</td>
<td>One or more Qualys top lists: Internal_10, External_10; multiple values are comma separated; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>cpe={value}</td>
<td>(Optional) One or more CPE values: Operating System, Application, Hardware, None; multiple values are comma separated.</td>
</tr>
<tr>
<td>qids_not_exploitable={0</td>
<td>1}</td>
</tr>
<tr>
<td>non_running_services={0</td>
<td>1}</td>
</tr>
<tr>
<td>sans_20={0</td>
<td>1}</td>
</tr>
<tr>
<td>nac_nam={0</td>
<td>1}</td>
</tr>
<tr>
<td>vuln_provider={value}</td>
<td>Provider of the vulnerability if not Qualys; valid value is iDefense</td>
</tr>
<tr>
<td>cvss_base={value}</td>
<td>CVSS base score value (matches greater than or equal to this value); to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>cvss_temp={value}</td>
<td>CVSS temporal score value (matches greater than or equal to this value); to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>cvss_access_vector={value}</td>
<td>CVSS access vector, one of: Undefined, Local, Adjacent_Network, Network; to unset value use update request and set to empty value</td>
</tr>
<tr>
<td>cvss_base_operand={value}</td>
<td>Set the value to 1 to use the greater than equal to operand. Set the value to 2 to use the less than operand. You must always specify the &quot;cvss_base&quot; parameter along with the &quot;cvss_base_operand&quot; parameter in the API request.</td>
</tr>
<tr>
<td>cvss_temp_operand={value}</td>
<td>Set the value to 1 to use the greater than equal to operand. Set the value to 2 to use the less than operand. You must always specify the &quot;cvss_temp&quot; parameter along with the &quot;cvss_temp_operand&quot; parameter in the API request.</td>
</tr>
</tbody>
</table>
### Parameter Value

**cvss3_base={value}**  
CVSS3 base score value assigned to the CVEs by NIST (matches greater than, less than, or equal to this value); to unset value use update request and set to empty value.

**cvss3_temp={value}**  
CVSS3 temporal score value assigned to the CVEs by NIST (matches greater than, less than, or equal to this value); to unset value use update request and set to empty value.

**cvss3_base_operand={value}**  
Set the value to 1 to use the greater than equal to operand. Set the value to 2 to use the less than operand. You must always specify the "cvss3_base" parameter along with the "cvss3_base_operand" parameter in the API request.

**cvss3_temp_operand={value}**  
Set the value to 1 to use the greater than equal to operand. Set the value to 2 to use the less than operand. You must always specify the "cvss3_temp" parameter along with the "cvss3_temp_operand" parameter in the API request.

### User modified filters
The user_modified* parameters are mutually exclusive, only one of these can be passed per request.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_modified_date_between={value}</td>
<td>date range in format (mm/dd/yyyy-mm/dd/yyyy)</td>
</tr>
<tr>
<td>user_modified_date_today={0</td>
<td>1}</td>
</tr>
<tr>
<td>user_modified_date_in_previous={value}</td>
<td>one of: Year, Month, Week, Quarter</td>
</tr>
<tr>
<td>user_modified_date_within_last_days={value}</td>
<td>number of days: 1-9999</td>
</tr>
<tr>
<td>not_user_modified={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

### Service modified filters
These parameters are mutually exclusive, only one of these can be passed per request.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>service_modified_date_between={value}</td>
<td>date range in format (mm/dd/yyyy-mm/dd/yyyy)</td>
</tr>
<tr>
<td>service_modified_date_today={0</td>
<td>1}</td>
</tr>
<tr>
<td>service_modified_date_in_previous={value}</td>
<td>one of: Year, Month, Week, Quarter</td>
</tr>
</tbody>
</table>
Published filters

These parameters are mutually exclusive, only one of these can be passed per request.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>service_modified_date_within_last_days=[value]</td>
<td>number of days: 1-9999</td>
</tr>
<tr>
<td>not_service_modified=[0</td>
<td>1]</td>
</tr>
</tbody>
</table>

Sample - Create dynamic search list

API request:
```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=create&title=My+Dynamic+Search+List&global=1&published_date_within_last_days=7&patch_available=1" "https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/dynamic/"
```

XML response:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2015-09-01T21:32:40Z</DATETIME>
    <TEXT>New search list created successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>136992</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Create dynamic search list, CVSS scores

API request:
Request for CVSS2 base scores: greater than equal to 3, CVSS 2 temporal scores less than 2, CVSS3 base scores greater than or equal to 2, CVSS3 temporal scores less than 2.

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo2" -d "action=create&title=mytest_DL313&cvss_base=3&cvss_base_operand=1&cvss_temp=2&cvss_temp_operand=2&cvss3_base=2&cvss3_base_operand=1&cvss3_temp=2&cvss3_temp_operand=2" "https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/dynamic/"

Update dynamic search list

Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=[0</td>
<td>1]</td>
</tr>
<tr>
<td>id=[id]</td>
<td>(Required) The ID of the search list you want to update.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Optional) The search list title. Maximum is 256 characters (ascii).</td>
</tr>
<tr>
<td>global=[0</td>
<td>1]</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) User defined comments.</td>
</tr>
<tr>
<td>[criteria]</td>
<td>(Optional) See “Search criteria”</td>
</tr>
<tr>
<td></td>
<td>Only criteria specified in an update request will overwrite existing criteria, if any. For example, if a search list has confirmed_severities=3,4 and you make an update request with confirmed_severities=5, the search list will be updated to confirmed_severities=5.</td>
</tr>
<tr>
<td>unset_user_modified_date=</td>
<td>(Optional) Set to empty value to unset the user modified date in the search list parameters.</td>
</tr>
<tr>
<td>unset_published_date=</td>
<td>(Optional) Set to empty value to unset the published date in the search list parameters.</td>
</tr>
<tr>
<td>unset_service_modified_date=</td>
<td>(Optional) Set to empty value to unset the service modified date in the search list parameters.</td>
</tr>
</tbody>
</table>

Sample - Update dynamic search list

API request:
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=update&id=136992" "https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/dynamic/"

XML response:
Delete dynamic search list

Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0</td>
<td>(Optional) Specify 1 to show input parameters in XML output.</td>
</tr>
<tr>
<td>id={id}</td>
<td>(Required) The ID of the search list you want to delete.</td>
</tr>
</tbody>
</table>

Sample - Delete dynamic search list

API request:

```bash
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=delete&id=123456" "https://qualysapi.qualys.com/api/2.0/fo/qid/search_list/dynamic/"
```

XML response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2015-09-01T21:32:40Z</DATETIME>
    <TEXT>search list deleted successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>123456</VALUE>
      </ITEM>
    </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```
Vendor IDs and References

/api/2.0/fo/vendor/?action=list_vendors
/api/2.0/fo/vendor/?action=list_vendor_references

List vendor IDs and names. This vendor information may be defined as part of dynamic search list query criteria.

Permissions - All users except Auditors have permission to run this API.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={value}</td>
<td>(Required) Set to &quot;list_vendors&quot; to list vendor IDs and names. Set to &quot;list_vendor_references&quot; to list vendor references for QIDs.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={id1,id2,...}</td>
<td>(Optional for action=list) One or more vendors IDs to list those vendors only.</td>
</tr>
<tr>
<td>qids={id1,id2,...}</td>
<td>(Optional for action=list_vendor_references) One or more QIDs to list vendors references for those QIDs only.</td>
</tr>
</tbody>
</table>

Sample - List vendor IDs and names

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/vendor/?action=list_vendors&ids=458,1967"
```

XML response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE VENDOR_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/vendor/vendor_list_output.dtd">
<VENDOR_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2015-09-02T09:23:52Z</DATETIME>
    <VENDORS>
      <VENDOR>
        <ID>458</ID>
        <NAME>
          <![CDATA[3com]]>
      </VENDOR>
    </VENDORS>
  </RESPONSE>
</VENDOR_LIST_OUTPUT>
```
Sample - List vendor references for qids

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/vendor/?action=list_vendor_re
ferences"

XML response:

<?xml version="1.0" encoding="UTF-8" ?>
<![DOCTYPE VENDOR_REFERENCE_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/vendor/vendor_reference_list_output.dtd">
<VENDOR_REFERENCE_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2015-09-02T09:27:34Z</DATETIME>
    <VENDOR_REFERENCES>
      <VENDOR_REFERENCE>
        <QID>195464</QID>
        <REFERENCE_INFO>
          <REFERENCE>
            <![CDATA[USN-2186-1]]>
          </REFERENCE>
          <URL>
          </URL>
        </REFERENCE_INFO>
      </VENDOR_REFERENCE>
      <VENDOR_REFERENCE>
        <QID>115844</QID>
        <REFERENCE_INFO>
          <REFERENCE>
            <![CDATA[RHSA-2008-0508]]>
          </REFERENCE>
          <URL>
            <![CDATA[http://rhn.redhat.com/errata/RHSA-2008-0508.html]]>
          </URL>
        </REFERENCE_INFO>
      </VENDOR_REFERENCE>
      <VENDOR_REFERENCE>
        <QID>151800</QID>
        <REFERENCE_INFO>
          <REFERENCE>
            <![CDATA[RHSA-2008-0519]]>
          </REFERENCE>
          <URL>
            <![CDATA[http://rhn.redhat.com/errata/RHSA-2008-0519.html]]>
          </URL>
        </REFERENCE_INFO>
      </VENDOR_REFERENCE>
    </VENDOR_REFERENCES>
    ... 
  </RESPONSE>
</VENDOR_REFERENCE_LIST_OUTPUT>
DTD
<platform API server>/api/2.0/fo/vendor/vendor_reference_list_output.dtd
Scan Authentication

Create, edit, list, delete authentication records for authenticated (trusted) scanning of various technologies (i.e. Windows, Unix, Docker, Oracle, etc).

Permissions
User Permissions Summary

List Auth Records
List Authentication Records
List Authentication Records by Type

Auth Record types
Application Server Records
- Apache, MIIS, IBM Websphere, Tomcat
Azure MS SQL Record
Docker Record
HTTP Record
IBM DB2 Record
InformixDB Record
JBoss Server record
Kubernetes Record
MariaDB Record
MongoDB Record
MS SQL Record
MySQL Record
Neo4j Record
Nginx Record
Oracle Record
Oracle Listener Record
Oracle WebLogic Server Record
Palo Alto Firewall Record

PostgreSQL Record
SAP Hana Record
SAP IQ Record
SNMP Record
Sybase Record
Unix Record
VMware Record
Windows Record
MS Exchange Server
Oracle HTTP Server Record
vCenter - ESXi Mapping Records
Network SSH Record
User Permissions Summary

A summary is provided below. For complete details, see “Managing Authentication Records” in Qualys online help.

Maximum Records per request

A maximum of 1,000 authentication records can be processed per request. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

View Record List

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>View all authentication records in subscription.</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>View authentication records which contain hosts in the user’s business unit.</td>
</tr>
<tr>
<td>Scanner</td>
<td>View authentication records which contain hosts in the user’s assigned asset groups.</td>
</tr>
<tr>
<td>Auditor, Reader</td>
<td>No permissions.</td>
</tr>
</tbody>
</table>

Create Record

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>Create authentication records for hosts in the subscription.</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>Create authentication records for hosts in the user’s business unit. The permission “create/edit authentication records” must be granted in the user’s account.</td>
</tr>
<tr>
<td>Auditor, Scanner, Reader</td>
<td>No permissions.</td>
</tr>
</tbody>
</table>

Update/Delete Record

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>Update and delete authentication records.</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>Update and delete authentication records. The permission “create/edit authentication records/vaults” must be granted in the user’s account. To edit a record, at least one host in the record must be in the user’s business unit. To delete a record, all hosts in the record must also be in the user’s business unit.</td>
</tr>
<tr>
<td>Auditor, Scanner, Reader</td>
<td>No permissions.</td>
</tr>
</tbody>
</table>
List Authentication Records

/api/2.0/fo/auth/?action=list

[GET]  [POST]

List all authentication records visible to the user for all technologies (i.e. Windows, Unix, Docker, etc).

A maximum of 1,000 authentication records can be processed per request. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=[0</td>
<td>1]</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Optional) Show only authentication records which have a certain string in the record title.</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) Show only authentication records which have a certain string in the record comments.</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Optional) Show only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(Optional) Show only authentication records which have a minimum ID value. A valid ID is required.</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(Optional) Show only authentication records which have a maximum ID value. A valid ID is required.</td>
</tr>
</tbody>
</table>

**DTD for list records**

<platform API server>/api/2.0/fo/auth/auth_records.dtd
Sample - List authentication records, multiple technologies

```xml
<AUTH_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-05-21T13:32:17Z</DATETIME>
    <AUTH_RECORDS>
      <AUTH_UNIX_RECORDS>
        <ID_SET>
          <ID_RANGE>17-41</ID_RANGE>
          <ID_RANGE>62-119</ID_RANGE>
        </ID_SET>
      </AUTH_UNIX_RECORDS>
      <AUTH_WINDOWS_RECORDS>
        <ID_SET>
          <ID_RANGE>1-6</ID_RANGE>
        </ID_SET>
      </AUTH_WINDOWS_RECORDS>
      <AUTH_ORACLE_RECORDS>
        <ID_SET>
          <ID>7</ID>
        </ID_SET>
      </AUTH_ORACLE_RECORDS>
      <AUTH_SNMP_RECORDS>
        <ID_SET>
          <ID>4114</ID>
          <ID_RANGE>4117-4121</ID_RANGE>
        </ID_SET>
      </AUTH_SNMP_RECORDS>
      <AUTH_IBM_DB2_RECORDS>
        <ID_SET>
          <ID>6</ID>
        </ID_SET>
      </AUTH_IBM_DB2_RECORDS>
    </AUTH_RECORDS>
  </RESPONSE>
</AUTH_LIST_OUTPUT>
```
List Authentication Records by Type

/api/2.0/fo/auth/<type>

{GET}  {POST}

List authentication records visible to the user for a specific technology (i.e. Unix, Windows, Docker, Sybase etc).

[type] will be a supported technology like: docker, http, ibm_db2, mongodb, ms_exchange, ms_sql, mysql, oracle, oracle_listener, oracle_weblogic, palo_alto_firwall, postgresql, snmp, sybase, unix (for Unix, Cisco, Checkpoint Firewall), network_ssh, vmware, windows. For application servers: apache, ms_iis, ibm_websphere, tomcat.

A maximum of 1,000 authentication records can be processed per request. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Optional) how only authentication records which have a certain string in the record title.</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) Show only authentication records which have a certain string in the record comments.</td>
</tr>
<tr>
<td>details={Basic</td>
<td>All</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Optional) Show only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(Optional) Show only authentication records which have a minimum ID value. A valid ID is required.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_max={value}</td>
<td>(Optional) Show only authentication records which have a maximum ID value. A valid ID is required.</td>
</tr>
</tbody>
</table>

#### Oracle Records

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>template_auth_id={value}</td>
<td>(Optional) Specify the template ID for an Oracle system record template to only show Oracle records associated with the specified template.</td>
</tr>
<tr>
<td>template_auth_name={value}</td>
<td>(Optional) Specify the template name for an Oracle system record template to only show Oracle records associated with the specified template.</td>
</tr>
<tr>
<td>is_template={0</td>
<td>1}</td>
</tr>
<tr>
<td>status={0</td>
<td>1}</td>
</tr>
<tr>
<td>is_system_created={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

#### MongoDB Records

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>template_auth_id={value}</td>
<td>(Optional) Specify the template ID for an MongoDB system record template to only show MongoDB records associated with the specified template.</td>
</tr>
<tr>
<td>template_auth_name={value}</td>
<td>(Optional) Specify the template name for an MongoDB system record template to only show MongoDB records associated with the specified template.</td>
</tr>
<tr>
<td>is_template={0</td>
<td>1}</td>
</tr>
<tr>
<td>status={0</td>
<td>1}</td>
</tr>
<tr>
<td>is_system_created={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

### DTD for list record type

```xml
<platform API server>/api/2.0/fo/auth/<type>/
```

where `<type>` is the authentication record type, such as unix, windows, oracle, etc.
Sample - List Unix and Cisco records

<AUTH_UNIX_LIST_OUTPUT>
<RESPONSE>
<DATETIME>2017-05-21T13:32:17Z</DATETIME>
<AUTH_UNIX_LIST>
<AUTH_UNIX>
<ID>678</ID>
<TITLE><![CDATA[My Ubuntu credentials]]></TITLE>
<UserName><![CDATA[bumbler]]></Username>
<ROOT_TOOL>Sudo</ROOT_TOOL>
<CLEARTEXT_PASSWORD>0</CLEARTEXT_PASSWORD>
<IP_SET>
<IP_RANGE>10.10.10.168-10.10.10.195</IP_RANGE>
</IP_SET>
<CREATED>
<DATETIME>2017-04-20T01:01:01</DATETIME>
<BY>quays_es11</BY>
</CREATED>
<LAST_MODIFIED>
<DATETIME>2017-04-20T01:01:01</DATETIME>
<BY>quays_es11</BY>
</LAST_MODIFIED>
</COMMENTS><![CDATA[Development lab]]></COMMENTS>
</AUTH_UNIX>
...
</AUTH_UNIX_LIST>
<WARNING_LIST>
<WARNING>
<CODE>1980</CODE>
<TEXT>1000 record limit exceeded. Use URL to get next batch of records.</TEXT>
</WARNING>
</WARNING_LIST>
<GLOSSARY>
<User_LIST>
<User>
<User_LOGIN>quays_es11</User_LOGIN>
<FIRST_NAME>Ernie</FIRST_NAME>
<LAST_NAME>Smith</LAST_NAME>
</User>
</USER_LIST>
</GLOSSARY>

<URL>https://qualysapi.qualys.com/api/2.0/fo/auth/?action=list&id_min=3457</URL>
</WARNING>
</WARNING_LIST>
</GLOSSARY>
Sample list Oracle record

This sample shows details for a single Oracle record specified by ID. The XML output identifies whether the record is system created, is active and is a template. In this example, the record listed is not system created. It is active and it is a template record.

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=list&ids=2237956" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_ORACLE_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/auth_oracle_list_output.dtd">
<AUTH_ORACLE_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-04-23T18:44:27Z</DATETIME>
    <AUTH_ORACLE_LIST>
      <AUTH_ORACLE>
        <ID>2237956</ID>
        <TITLE><![CDATA[OracleRecordTemplate]]></TITLE>
        <USERNAME><![CDATA[OracleUser]]></USERNAME>
        <CREATED>
          <DATETIME>2020-04-23T18:43:59Z</DATETIME>
          <BY>rey_pt11</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2020-04-23T18:43:59Z</DATETIME>
        </LAST_MODIFIED>
        <IS_SYSTEM_CREATED>0</IS_SYSTEM_CREATED>
        <IS_ACTIVE>1</IS_ACTIVE>
        <IS_TEMPLATE>1</IS_TEMPLATE>
        <COMMENTS><![CDATA[my comments]]></COMMENTS>
      </AUTH_ORACLE>
    </AUTH_ORACLE_LIST>
  </RESPONSE>
</AUTH_ORACLE_LIST_OUTPUT>

Sample list MongoDB record

This sample shows details for a single MongoDB record specified by ID. The XML output identifies whether the record is system created, is active and is a template. In this example, the record listed is not system created. It is active and it is a template record.

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=list&ids=2237956" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_ORACLE_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/auth_oracle_list_output.dtd">
<AUTH_ORACLE_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-04-23T18:44:27Z</DATETIME>
    <AUTH_ORACLE_LIST>
      <AUTH_ORACLE>
        <ID>2237956</ID>
        <TITLE><![CDATA[OracleRecordTemplate]]></TITLE>
        <USERNAME><![CDATA[OracleUser]]></USERNAME>
        <CREATED>
          <DATETIME>2020-04-23T18:43:59Z</DATETIME>
          <BY>rey_pt11</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2020-04-23T18:43:59Z</DATETIME>
        </LAST_MODIFIED>
        <IS_SYSTEM_CREATED>0</IS_SYSTEM_CREATED>
        <IS_ACTIVE>1</IS_ACTIVE>
        <IS_TEMPLATE>1</IS_TEMPLATE>
        <COMMENTS><![CDATA[my comments]]></COMMENTS>
      </AUTH_ORACLE>
    </AUTH_ORACLE_LIST>
  </RESPONSE>
</AUTH_ORACLE_LIST_OUTPUT>
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=list&ids=6847704" "https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/"

**XML Output**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_MONGODB_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/auth_mongodb_list_output.dtd">
<AUTH_MONGODB_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2023-01-05T12:47:03Z</DATETIME>
    <AUTH_MONGODB_LIST>
      <AUTH_MONGODB>
        <ID>6847704</ID>
        <TITLE><![CDATA[MongoDB system record template]]></TITLE>
        <USERNAME><![CDATA[root]]></USERNAME>
        <CREDENTIAL_TYPE><![CDATA[local]]></CREDENTIAL_TYPE>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <REQUIRE_CERT><![CDATA[0]]></REQUIRE_CERT>
        <CREATED>
          <DATETIME>2023-01-05T07:16:46Z</DATETIME>
          <BY>vt_sm1</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2023-01-05T07:16:46Z</DATETIME>
        </LAST_MODIFIED>
        <IS_SYSTEM_CREATED>0</IS_SYSTEM_CREATED>
        <IS_ACTIVE>1</IS_ACTIVE>
        <IS_TEMPLATE>1</IS_TEMPLATE>
      </AUTH_MONGODB>
    </AUTH_MONGODB_LIST>
  </RESPONSE>
</AUTH_MONGODB_LIST_OUTPUT>
```
**Application Server Records**

/\api/2.0/fo/auth/\{(web app server)/\}

where {web app server} is one of `apache, ms_iis, ibm_websphere, tomcat`

[POST]

Create, update, list and delete application server records for authenticated scans of web application servers. Application Server records are used to authenticate to various web app servers.

Instance discovery and auto record creation is supported for Apache Web Server, IBM WebSphere, JBoss, Tomcat, Oracle and MongoDB. Learn more about instance discovery and auto record creation in online help (log in to your Qualys account, go to Help > Online Help and search for System Authentication Records).

**Supported servers**

<table>
<thead>
<tr>
<th>API URL (/api/2.0/fo...)</th>
<th>Supported Versions</th>
</tr>
</thead>
</table>
| /auth/apache/             | - Apache HTTP Server 2.2 and 2.4  
- IBM HTTP Server 7.x, 8.x and 9.x  
- VMware vFabric Web Server 5.x  
- Pivotal Web Server 6.x  
Compliance scans are supported (using PC) |
| /auth/ms_iis/             | MS IIS 6.0, 7.x, 8.x and 10 for Windows  
Compliance scans are supported (using PC) |
| /auth/ibm_websphere/     | IBM WebSphere Application Server 7.x, 8.x and 9.x  
Compliance scans are supported (using PC) |
| /auth/tomcat/             | Windows:  
- Apache Tomcat 7.x, 8.x and 9.x  
Unix:  
- Apache Tomcat 6.x, 7.x, 8.x and 9.x  
- VMware vFabric tc Server 2.9.x  
- Pivotal tc Server 3.x  
Vulnerability and Compliance scans are supported (using VM, PC) |
### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required for create) The title of the Server record. The title must be unique and may include a maximum of 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) User defined notes about the Server record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as &lt;script&gt;) cannot be included; if tags are included an error is returned and the request fails.</td>
</tr>
</tbody>
</table>

### Application Server

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unix_apache_config_file={value}</td>
<td>(Required to create an Apache Web Server record; valid only for this record). The path to the Apache configuration file.</td>
</tr>
<tr>
<td>unix_apache_control_command={value}</td>
<td>(Required to create an Apache Web Server record; valid only for this record) The path to the Apache control command. For IBM HTTP Server, enter the path to the IBM HTTP Server &quot;bin&quot; directory or the specific location of &quot;apachectl&quot;. For VMware vFabric Web Server, enter the path to the VMware vFabric global &quot;bin&quot; directory or the specific location of &quot;httpdctl&quot; for a web server instance.</td>
</tr>
<tr>
<td>windows_apache_config_file={value}</td>
<td>(Required to create Apache HTTP and IBM HTTP server records; valid only for this record). The Windows path to the Apache HTTP and IBM HTTP server configuration file.</td>
</tr>
<tr>
<td>windows_apache_control_command={value}</td>
<td>(Required to create Apache HTTP and IBM HTTP server records; valid only for this record) The Windows path to the Apache HTTP and IBM HTTP server control command. For IBM HTTP Server, enter the path to the IBM HTTP Server &quot;bin&quot; directory or the specific location of &quot;apachectl&quot;.</td>
</tr>
<tr>
<td>unix_installation_dir={value}</td>
<td>(Required to create an IBM WebSphere App Server record; valid only for this record) The directory where the WebSphere application is installed.</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>unix_dir_mode={value}</td>
<td>(Optional for IBM WebSphere App Server record; valid only for this record) Specify the Unix directory mode. Valid values are installation_dir (for installation directory) and server_dir (for server directory). When not specified, installation_dir is used.</td>
</tr>
<tr>
<td>windows_installation_dir={value}</td>
<td>(Required to create an IBM WebSphere App Server record; valid only for this record) The Windows directory where the WebSphere application is installed.</td>
</tr>
</tbody>
</table>
| installation_path={value} | (Required to create Tomcat Server record on UNIX; valid only for this record) The directory where the tomcat server is installed.  
Examples:  
/opt/apache-tomcat-7.0.57 (e.g. $CATALINA_HOME)  
/opt/vmware/vfabric-tc-server-standard  
/opt/pivotal/pivotal-tc-server-standard |
| instance_path={value} | (Optional to create or update Tomcat Server record on UNIX; valid only for this record) The directory where the tomcat server instance(s) are installed. You can specify a single tomcat instance (use with auto_discover_instances=0), or multiple instances (use with auto_discover_instances=1). Leave unspecified when the instance directory is the same as the installation directory or when your targets have different types of tomcat servers.  
Examples:  
/opt/apache-tomcat-7.0.57 (e.g. $CATALINA_BASE)  
/opt/vmware/vfabric-tc-server-standard/tc1  
/opt/pivotal/pivotal-tc-server-standard/tc1 |
| auto_discover_instances={0|1} | (Optional to create or update Tomcat Server record; valid only for this record) Specify auto_discover_instances=1 and we’ll find all tomcat server instances for you. Applies to VMware vFabric and Pivotal when you’ve specified a directory with multiple instances or you did not specify an instance.  
When unspecified (auto_discover_instances=0), we will not auto discover instances. Applies to Apache Tomcat or when you’ve specified a single instance. |
| installation_path_windows | (Required to create Tomcat Server record on Windows; valid only for this record) The directory where the tomcat server is installed.  
Example: C:\tomcat\apache-tomcat-8.5.11 |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_path_windows</td>
<td>(Optional to create or update Tomcat Server record on UNIX; valid only for this record) The directory where the tomcat server instance(s) are installed.</td>
</tr>
<tr>
<td></td>
<td>Example: C:\tomcat\apache-tomcat-8.5.11</td>
</tr>
<tr>
<td>service_name</td>
<td>(Optional) The Tomcat service name that identifies the server instance to be authenticated to.</td>
</tr>
<tr>
<td>Apache Server only</td>
<td></td>
</tr>
<tr>
<td>status={0</td>
<td>1}</td>
</tr>
<tr>
<td></td>
<td>For list request (action set to list) - By default active and inactive auth records are listed. Set to 0 to list only inactive records or set to 1 to list only active records.</td>
</tr>
<tr>
<td></td>
<td>For create/update request (action set to create or update) - By default a new record is set to active (1). Set to 0 for inactive record, or 1 for active record. For update action, this parameter is valid only when user created records are specified in the request.</td>
</tr>
<tr>
<td>is_system_created={0</td>
<td>1}</td>
</tr>
<tr>
<td>Target Hosts</td>
<td></td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required to create record) Add IP addresses of the hosts you want to scan using this record.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional and valid only to update record) Add IP address(es) to the IP list for an existing record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional and valid only to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>

**Sample - Create Apache record**

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=create&title=Apache+Record&unix_apache_config_file=/opt/IBM/HTTPServer/conf/httpd.conf1&unix_apache_control_command=/opt/IBM/HTTPServer/bin2&ips=10.10.25.25" "https://qualysapi.qualys.com/api/2.0/fo/auth/apache/
```
Sample - Update Apache record


DTDs for server records

<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/apache/auth_apache_list_output.dtd
<platform API server>/api/2.0/fo/auth/ms_iis/auth_ms_iis_list_output.dtd
<platform API server>/api/2.0/fo/auth/ibm_websphere/auth_ibm_websphere_list_output.dtd
<platform API server>/api/2.0/fo/auth/tomcat/auth_tomcat_list_output.dtd
Azure MS SQL Record

/api/2.0/fo/auth/azure_ms_sql/

[POST]

Create, update, list, and delete Azure MS SQL records for compliance scans (using PC). Compliance scans are supported (using PC).

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) User defined comments. Maximum of 1999 characters.</td>
</tr>
</tbody>
</table>

**Login credentials**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>provider_name={value}</td>
<td>(Optional) Name of the cloud service provider. The only value supported is azure. This value will be passed by default.</td>
</tr>
<tr>
<td>login_type={basic</td>
<td>vault}</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The username to be used for authentication to Azure MS SQL. The username must contain '@'.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required to create record, optional to update record) when login_type=basic, specify the password to be used for authentication to Azure MS SQL. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td>instance_name={value}</td>
<td>(Optional to create or update record) The name of the database instance to be scanned. This is the instance name assigned to the TCP/IP port. Important: This is not the host name that is assigned to the Azure MS SQL Server instance name. The only value supported is MSSQLSERVER. This value will be passed by default. Currently, we do not support named instances for this parameter.</td>
</tr>
</tbody>
</table>
### Parameter |
<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>database_name={value}</strong></td>
</tr>
<tr>
<td>(Optional to create or update record) The database name of the Azure MS SQL database to be scanned. The database name may contain a maximum of 128 characters. These parameters are mutually exclusive: database_name and auto_discover_databases=1.</td>
</tr>
<tr>
<td>**auto_discover_databases={0</td>
</tr>
<tr>
<td>(Optional to create or update record) The database name of the Azure MS SQL database to be scanned. The database name may contain a maximum of 128 characters. These parameters are mutually exclusive: database_name and auto_discover_databases=1.</td>
</tr>
<tr>
<td><strong>port={value}</strong></td>
</tr>
<tr>
<td>(Required to create record, optional to update record) The port number assigned to the database instance to be scanned.</td>
</tr>
</tbody>
</table>

### Vaults |
| **vault_type={value}** |
| (Required to create record when login_type=vault) The third party vault to be used to retrieve the password for login. Certain vaults support this capability. See Vault Support matrix. |
| **vault_id={value}** |
| (Required only when action=create and login_type=vault) The ID of the vault you want to use. |
| **{vault parameters}** |
| (Required only when action=create and login_type=vault) Vault specific parameters required depend on the vault type you’ve selected. See Vault Definition. |

### Target Hosts |
| **ips={value}** |
| (Required to create record) IPs to be added to your Azure MS SQL Record record. You may enter a combination of IPs and IP ranges to identify compliance hosts. Multiple entries are comma separated. |
| **add_ips={value}** |
| (Optional and valid only to update record) Add IPs to the IPs list for this record. Multiple IPs/ranges are comma separated. |
| **remove_ips={value}** |
| (Optional and valid to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated. This parameter and the ips parameter cannot be specified in the same request. |
| **network_id={value}** |
| (Optional and valid when the networks feature is enabled) The network ID for the record. |
Sample - List Azure MS SQL Records

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list&ids=4620763"
"https://qualysapi.qualys.com/api/2.0/fo/auth/azure_ms_sql/
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_AZURE_MS_SQL_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/azure_ms_sql/dtd/auth_list_output.dtd">
<AUTH_AZURE_MS_SQL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-04-23T13:53:08Z</DATETIME>
    <AUTH_AZURE_MS_SQL_LIST>
      <AUTH_AZURE_MS_SQL>
        <ID>4620763</ID>
        <TITLE><![CDATA[AzureMSSQL_Auth_API]]></TITLE>
        <PROVIDER_NAME><![CDATA[Azure]]></PROVIDER_NAME>
        <USERNAME><![CDATA[john_user@qualys.com]]></USERNAME>
        <INSTANCE><![CDATA[MSSQLSERVER]]></INSTANCE>
        <DATABASE><![CDATA[testdb]]></DATABASE>
        <PORT>42</PORT>
        <IP_SET>
          <IP>1.1.1.4</IP>
        </IP_SET>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <CREATED>
          <DATETIME>2021-04-01T11:47:51Z</DATETIME>
          <BY>up_at</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2021-04-01T11:47:51Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_AZURE_MS_SQL>
    </AUTH_AZURE_MS_SQL_LIST>
  </RESPONSE>
</AUTH_AZURE_MS_SQL_LIST_OUTPUT>
```

Sample - Create Azure MS SQL Record (with basic login)

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=create&title=my-azuremssql-record&ips=1.1.1.4&port=42&database_name=dbname"
"https://qualysapi.qualys.com/api/2.0/fo/auth/azure_ms_sql/
```
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
  "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-04-23T11:47:51Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>4620763</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Update Azure MS SQL Record (with auto_discover_databases=1)

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
  "action=update&auto_discover_databases=1&ids=207024"
  "https://qualysapi.qualys.com/api/2.0/fo/auth/azure_ms_sql/
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
  "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>207024</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Delete Azure MS SQL Records

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
  "action=delete&ids=4620768"
  "https://qualysapi.qualys.com/api/2.0/fo/auth/azure_ms_sql/
```

**Response:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
```
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-04-26T13:12:51Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>4620768</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

**DTDs for auth type "azure_ms_sql"**

- <platform API server>/api/2.0/batch_return.dtd
- <platform API server>/api/2.0/fo/auth/azure_ms_sql/dtd/auth_list_output.dtd
### Cisco APIC 4.x Authentication Record

/`api/2.0/fo/auth/cisco_apic/`

**[POST]**

Create, update, and list, Cisco APIC 4.x authentication records, for the PC/SCA modules.

#### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>Required</td>
<td>String</td>
<td>Specify to list, create, update, delete, authentication records.</td>
</tr>
<tr>
<td>ids</td>
<td>Required</td>
<td>Integer</td>
<td>Specify a single or comma separated valid Cisco APIC type authentication record ID(s).</td>
</tr>
<tr>
<td>title</td>
<td>Required</td>
<td>Alpha-Numeric</td>
<td>Specify the title for the authentication record.</td>
</tr>
<tr>
<td>ips</td>
<td>Required</td>
<td>Integer</td>
<td>Specify the IP address(es) the server will log into using the record's credentials. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>username</td>
<td>Required</td>
<td>String</td>
<td>Specify username for authentication login.</td>
</tr>
<tr>
<td>password</td>
<td>Required</td>
<td>String</td>
<td>Specify password for authentication login.</td>
</tr>
<tr>
<td>port</td>
<td>Required</td>
<td>Integer</td>
<td>Specify port number required on Cisco APIC devices.</td>
</tr>
<tr>
<td>ssl_verify</td>
<td>Optional</td>
<td>Integer</td>
<td>SSL verification is skipped by default. Set to 1 if you want to verify the server's certificate is valid and trusted.</td>
</tr>
</tbody>
</table>

**Require Certificate**

Optional

Alpha-Numeric

It contains two textfields, certificate and privatekey.

**Vault Parameter**

**Note:** The vault parameters are supported on the following vault types (Secret server Vault, Quest Vault, Hashicorp Vault, CyberArk AIM Vault, CyberArk PIM Vault, and AzureKey Vault).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vault_type</td>
<td>Required</td>
<td>String</td>
<td>Specify if create, and login_type=vault. (private key only supports &quot;Cyber-Ark AIM&quot; and &quot;BeyondTrust PBPS&quot;, whereas passphrase does not support &quot;BeyondTrust PBPS&quot;).</td>
</tr>
<tr>
<td>vault_id</td>
<td>Required</td>
<td>Integer</td>
<td>Specify if create, and login_type=vault. The ID of the vault to be used to retrieve the password for login.</td>
</tr>
<tr>
<td>file</td>
<td>Required</td>
<td>String</td>
<td>Specify to create, if vault_type= &quot;Cyber-Ark AIM&quot; or &quot;Cyber-ARK PIM Suite&quot;.</td>
</tr>
</tbody>
</table>
Sample - List Cisco APIC auth records API

API Request:

curl -u "<token>" -H "X-Requested-With: curl" -d "action=list" "http://qualys_base_url/api/2.0/fo/auth/cisco_apic/

XML Response:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_CISCO_APIC_LIST_OUTPUT SYSTEM
"https://qualys_base_url/api/2.0/fo/auth/cisco_apic/auth_cisco_apic_list_output.dtd">
<AUTH_CISCO_APIC_LIST_OUTPUT>
  <RESPONSE>
    <AUTH_CISCO_APIC_LIST>
      <AUTH_CISCO_APIC>
        <ID>7850382</ID>
        <TITLE><![CDATA[user 1]]></TITLE>
        <USERNAME><![CDATA[user 1]]></USERNAME>
        <PORT>444</PORT>
        <WINDOWS_DOMAIN><![CDATA[aaaa]]></WINDOWS_DOMAIN>
        <SSL_VERIFY><![CDATA[1]]></SSL_VERIFY>
        <IP_SET>
          <IP>3.3.3.3</IP>
        </IP_SET>
        <REQUIRE_CERT><![CDATA[1]]></REQUIRE_CERT>
        <DOMAIN><![CDATA[aaaa]]></DOMAIN>
        <PORT><![CDATA[444]]></PORT>
        <SSL_VERIFY_WITH_HOST><![CDATA[1]]></SSL_VERIFY_WITH_HOST>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <CREATED>
          <BY>vt_sm1</BY>
        </CREATED>
      </AUTH_CISCO_APIC>
    </AUTH_CISCO_APIC_LIST>
  </RESPONSE>
</AUTH_CISCO_APIC_LIST_OUTPUT>

Parameter | Required/Optional | Data Type | Description
---|---|---|---
folder | Required | String | Specify to create, if vault_type="Cyber-Ark AIM" or "Cyber-ARK PIM Suite.
secret_name | Required | String | Specify to create, if vault_type="Thycotic Secret Server".
system_name | Required | String | Specify if yes(Quest) / no(BeyondTrust PBPS) (create, and vault_type="Quest Vault" or "BeyondTrust PBPS").
account_name | Required | String | Specify to create, if vault_type="BeyondTrust PBPS".
DTD Output:

DTD: <qualys_base_url>/api/2.0/fo/auth/auth_cisco_apic_list_output.dtd

<!-- QUALYS AUTH_RECORDS_OUTPUT DTD -->
<!ELEMENT AUTH_RECORDS_OUTPUT (REQUEST?, RESPONSE)>  

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>  
<!ELEMENT USER_LOGIN (#PCDATA)>  
<!ELEMENT RESOURCE (#PCDATA)>  
<!ELEMENT PARAM_LIST (PARAM+)>  
<!ELEMENT PARAM (KEY, VALUE)>  
<!ELEMENT KEY (#PCDATA)>  
<!ELEMENT VALUE (#PCDATA)>  
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>  

<!ELEMENT RESPONSE (DATETIME, AUTH_RECORDS?, WARNING_LIST?)>
<!ELEMENT AUTH_RECORDS (AUTH_UNIX_IDS?, AUTH_WINDOWS_IDS?,  
AUTH_ORACLE_IDS?, AUTH_ORACLE_LISTENER_IDS?, AUTH_SNMP_IDS?,  
AUTH_MS_SQL_IDS?, AUTH_IBM_DB2_IDS?, AUTH_VMWARE_IDS?, AUTH_MS_IIS_IDS?,  
AUTH_APACHE_IDS?, AUTH_IBM_WEBSPHERE_IDS?, AUTH_HTTP_IDS?,  
AUTH_SYBASE_IDS?, AUTH_MYSQL_IDS?, AUTH_TOMCAT_IDS?,  
AUTH_ORACLE_WEBLOGIC_IDS?, AUTH_DOCKER_IDS?, AUTH_POSTGRESQL_IDS?,  
AUTH_MONGODB_IDS?, AUTH_PALO_ALTO_FIREWALL_IDS?, AUTH_VCENTER_IDS?,  
AUTH_JBOSS_IDS?, AUTH_MARIADB_IDS?, AUTH_INFORMIXDB_IDS?,  
AUTH_MS_EXCHANGE_IDS?, AUTH_ORACLE_HTTP_SERVER_IDS?, AUTH_GREENPLUM_IDS?,  
AUTH_MICROSOFT_SHAREPOINT_IDS?, AUTH_KUBERNETES_IDS?,  
AUTH_SAPIQ_IDS?, AUTH_SAP_HANA_IDS?, AUTH_NEO4J_IDS?,  
AUTH_MICROSOFT_MS_SQL_IDS?, AUTH_NETWORK_SSH_IDS?, AUTH_NGINX_IDS?,  
AUTH_INFOBLOX_IDS?, AUTH_CISCO_APIC_IDS?)>
<!ELEMENT AUTH_UNIX_IDS (ID_SET)>
<!ELEMENT AUTH_WINDOWS_IDS (ID_SET)>
<!ELEMENT AUTH_ORACLE_IDS (ID_SET)>
<!ELEMENT AUTH_ORACLE_LISTENER_IDS (ID_SET)>
<!ELEMENT AUTH_SNMP_IDS (ID_SET)>
<!ELEMENT AUTH_MS_SQL_IDS (ID_SET)>
<!ELEMENT AUTH_IBM,DB2_IDS (ID_SET)>
<!ELEMENT AUTH_VMWARE_IDS (ID_SET)>
<!ELEMENT AUTH_MS_IIS_IDS (ID_SET)>
<!ELEMENT AUTH_APACHE_IDS (ID_SET)>
<!ELEMENT AUTH_IBM_WEBSPHERE_IDS (ID_SET)>
<!ELEMENT AUTH_HTTP_IDS (ID_SET)>
<!ELEMENT AUTH_SYBASE_IDS (ID_SET)>
<!ELEMENT AUTH_MYSQL_IDS (ID_SET)>
<!ELEMENT AUTH_TOMCAT_IDS (ID_SET)>
<!ELEMENT AUTH_ORACLE_WEBLOGIC_IDS (ID_SET)>
<!ELEMENT AUTH_DOCKER_IDS (ID_SET)>
<!ELEMENT AUTH_POSTGRESQL_IDS (ID_SET)>

Scan Authentication
Cisco APIC 4.x Authentication Record
Sample - Create Cisco APIC auth records Request API

**API Request:**

curl -u "<token>" -H "X-Requested-With: curl" -d "action=create&ips=3.3.3.3&title=testciscoapic&username=admin&password=abc123&port=443" "http://qualys_base_url/api/2.0/fo/auth/cisco_apic/

**XML Response:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualys_base_url/api/2.0/batch_return.dtd">
<BATCH_RETURN>
    <RESPONSE>
        <DATETIME>2023-08-09T11:03:04Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Created</TEXT>
                <ID_SET>
                    <ID>7850988</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```
Scan Authentication
Cisco APIC 4.x Authentication Record

Sample - Update Cisco APIC auth records Request API

API Request:
```
curl -u "<token>" -H "X-Requested-With: curl" -d "action=update&ids=7850382&title=testciscoapic"
"http://qualys_base_url/api/2.0/fo/auth/cisco_apic/
```

XML Response:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualys_base_url/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2023-08-09T10:48:33Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>7850382</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

DTD Output:

DTD: <qualys_base_url>/api/2.0/fo/auth/auth_cisco_apic_list_output.dtd

```
<!DOCTYPE auth_cisco_apic_list_output SYSTEM "https://qualys_base_url/api/2.0/batch_return.dtd">
<!ELEMENT auth_cisco_apic_list_output (request?, response?)>
<!ELEMENT request (datetime, user_login, resource, param_list?, post_data?)>
<!ELEMENT datetime (#pcdata)>
<!ELEMENT user_login (#pcdata)>
<!ELEMENT resource (#pcdata)>
<!ELEMENT param_list (param+)>
<!ELEMENT param (key, value)>
<!ELEMENT key (#pcdata)>
<!ELEMENT value (#pcdata)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT post_data (#pcdata)>
```
<!ELEMENT ID_RANGE (#PCDATA)>
<!ELEMENT GLOSSARY (USER_LIST?)>
<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>

<!-- EOF -->

**DTDs for auth type “cisco_apic”**

<platform API server>/api/2.0/batch_return.dtd

<platform API server>/api/2.0/fo/auth/auth_cisco_apic_list_output.dtd
DNS BIND Authentication Record

/api/2.0/fo/auth/bind/  
[POST]

Create, update, and list DNS BIND authentication records, for the PC/SCA modules.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>Required</td>
<td>String</td>
<td>Specify to list, create, update, delete, authentication records.</td>
</tr>
<tr>
<td>ids</td>
<td>Required</td>
<td>Integer</td>
<td>Specify a single or comma separated valid DNS BIND type authentication record ID(s).</td>
</tr>
<tr>
<td>title</td>
<td>Required</td>
<td>Alpha-Numeric</td>
<td>Specify the title for the authentication record.</td>
</tr>
<tr>
<td>ips</td>
<td>Required</td>
<td>Integer</td>
<td>Specify the IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>unix_bin_path</td>
<td>Required</td>
<td>Path</td>
<td>Specify absolute path of the DNS BIND Base64 encoded binary file location. Base64 encoded binary file path. Example- &quot;/usr/sbin/named&quot;.</td>
</tr>
<tr>
<td>unix_conf_path</td>
<td>Required</td>
<td>Path</td>
<td>Specify absolute path of the DNS BIND Base64 encoded configuration file path. Example- &quot;/etc/named.conf&quot;.</td>
</tr>
<tr>
<td>base_directory</td>
<td>Optional</td>
<td>Path</td>
<td>Specify Base64 encoded base directory. In BIND configuration file, if an include file is relative path, it is relative to this base_dir. Optional field, if not present, it will be derived from conf_path. Example- If conf_path is /etc/named.conf and if base_dir is not specified, then base_dir is set to &quot;/etc&quot;. It must be absolute path if specified.</td>
</tr>
<tr>
<td>chroot_directory</td>
<td>Optional</td>
<td>Path</td>
<td>Specify Base64 encoded chroot directory. Optional field, only needed if BIND runs in a self contained environment. If present, must be absolute path and it will be prefixed to all other 3 paths. Example- If chroot_dir is &quot;/var/bind&quot; and if bin_path is &quot;/usr/sbin/named&quot;, then the final bin_path will be &quot;/var/bind/usr/sbin/named&quot;.</td>
</tr>
</tbody>
</table>

Sample - List DNS BIND auth records API

API Request:

```c
curl -d "action=list" "http://qualys_base_url/api/2.0/fo/auth/bind/"
```
XML Response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_BIND_LIST_OUTPUT SYSTEM "https://qualys_base_url/api/2.0/fo/auth/bind/auth_bind_list_output.dtd">
<AUTH_BIND_LIST_OUTPUT>
  <AUTH_BIND_LIST>
    <AUTH_BIND>
      <ID>7882778</ID>
      <TITLE><![CDATA[user_1]]></TITLE>
      <IP_SET>
        <IP>40.40.40.6</IP>
      </IP_SET>
      <UNIX_BIN_PATH><![CDATA[/usr/bin/bind]]></UNIX_BIN_PATH>
      <UNIX_CONF_PATH><![CDATA[/etc/bind/bind.conf]]></UNIX_CONF_PATH>
      <BASE_DIRECTORY><![CDATA[/etc/bind]]></BASE_DIRECTORY>
      <CHROOT_DIRECTORY><![CDATA[/etc/bind]]></CHROOT_DIRECTORY>
      <CREATED>
        <DATETIME>2023-08-14T16:15:57Z</DATETIME>
        <BY>vt_sm1</BY>
      </CREATED>
      <LAST_MODIFIED>
        <DATETIME>2023-08-14T16:15:57Z</DATETIME>
      </LAST_MODIFIED>
    </AUTH_BIND>
  </AUTH_BIND_LIST>
</RESPONSE>
</AUTH_BIND_LIST_OUTPUT>
```

Sample - Create DNS BIND auth records Request API

API Request:

```
curl -d "action=create&ips=40.40.40.6&title=user_1bind2c&unix_bin_path=/etc&unix_conf_path=/etc/bind" "http://qualys_base_url:48443/api/2.0/fo/auth/bind/
```

XML Response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualys_base_url/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <BATCH_LIST>
    <BATCH>
      <TEXT>Successfully Created</TEXT>
      <ID_SET>
        <ID>7882779</ID>
      </ID_SET>
    </BATCH>
  </BATCH_LIST>
</RESPONSE>
</BATCH_RETURN>
```
Sample - Update DNS BIND auth records Request API

API Request:
```
curl -d "action=update&ids=7882778&title=testbind" 
"http://qualys_base_url:48443/api/2.0/fo/auth/bind/
```

XML Response:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualys_base_url/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <BATCH_LIST>
    <BATCH>
      <TEXT>Successfully Updated</TEXT>
      <ID_SET>
        <ID>7882778</ID>
      </ID_SET>
    </BATCH>
  </BATCH_LIST>
</RESPONSE>
</BATCH_RETURN>
```

DTD Output:

Auth Record list output DTD
```
<!-- QUALYS AUTH_RECORDS_OUTPUT DTD -->
<!ELEMENT AUTH_RECORDS_OUTPUT (REQUEST?, RESPONSE)>  
<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)>  
<!ELEMENT DATETIME (#PCDATA)>  
<!ELEMENT USER_LOGIN (#PCDATA)>  
<!ELEMENT RESOURCE (#PCDATA)>  
<!ELEMENT PARAM_LIST (PARAM+)>  
<!ELEMENT PARAM (KEY, VALUE)>  
<!ELEMENT KEY (#PCDATA)>  
<!ELEMENT VALUE (#PCDATA)>  
<!ELEMENT POST_DATA (#PCDATA)>  
<!ELEMENT RESPONSE (DATETIME, AUTH_RECORDS?, WARNING_LIST?)>  
```

<!ELEMENT AUTH_UNIX_IDS (ID_SET)>  
<!ELEMENT AUTH_WINDOWS_IDS (ID_SET)>  
<!ELEMENT AUTH_ORACLE_IDS (ID_SET)>  
<!ELEMENT AUTH_ORACLE_LISTENER_IDS (ID_SET)>  
<!ELEMENT AUTH_SNMP_IDS (ID_SET)>  
<!ELEMENT AUTH_MS_SQL_IDS (ID_SET)>  
<!ELEMENT AUTH_IBMDB2_IDS (ID_SET)>  
<!ELEMENT AUTH_VMWARE_IDS (ID_SET)>  
<!ELEMENT AUTH_MS_IIS_IDS (ID_SET)>  
<!ELEMENT AUTH_AZURE_MS_SQL_IDS (ID_SET)>  
<!ELEMENT AUTH_NETWORK_SSH_IDS (ID_SET)>  
<!ELEMENT AUTH_NGINX_IDS (ID_SET)>  
<!ELEMENT AUTH_INFOBLOX_IDS (ID_SET)>  
<!ELEMENT AUTH_BIND_IDS (ID_SET)>  
<!ELEMENT AUTH_SAPIQ_IDS (ID_SET)>  
<!ELEMENT AUTH_SAP_HANA_IDS (ID_SET)>  
<!ELEMENT AUTH_NEO4J_IDS (ID_SET)>  
<!ELEMENT AUTH_AZURE_MS_SQL_IDS (ID_SET)>  
<!ELEMENT AUTH_NETWORK_SSH_IDS (ID_SET)>  
<!ELEMENT AUTH_NGINX_IDS (ID_SET)>  
<!ELEMENT AUTH_INFOBLOX_IDS (ID_SET)>  
<!ELEMENT AUTH_BIND_IDS (ID_SET)>  

<!ELEMENT WARNING_LIST (WARNING+)>  
<!ELEMENT WARNING (CODE?, TEXT, URL?, ID_SET?)>  
<!ELEMENT CODE (#PCDATA)>  
<!ELEMENT TEXT (#PCDATA)>  
<!ELEMENT URL (#PCDATA)>  

<!ELEMENT ID_SET (ID|ID_RANGE)+>  
<!ELEMENT ID (#PCDATA)>  
<!ELEMENT ID_RANGE (#PCDATA)>  

<!-- EOF -->
DTD: <qualys_base_url>/api/2.0/fo/auth/auth_bind_list_output.dtd

Auth Infoblox list output DTD

<!-- QUALYS AUTH_BIND_LIST_OUTPUT DTD -->
<!ELEMENT AUTH_BIND_LIST_OUTPUT (REQUEST?, RESPONSE)> 

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, (AUTH_BIND_LIST|ID_SET)?, WARNING_LIST?, GLOSSARY?)>
<!ELEMENT AUTH_BIND_LIST (AUTH_BIND+)> 

<!ELEMENT AUTH_BIND (ID, TITLE, IP_SET?, UNIX_BIN_PATH?, UNIX_CONF_PATH?, BASE_DIRECTORY?, CHROOT_DIRECTORY?, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT IP_SET (IP|IP_RANGE)+>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>
<!ELEMENT UNIX_BIN_PATH (#PCDATA)>
<!ELEMENT UNIX_CONF_PATH (#PCDATA)>
<!ELEMENT BASE_DIRECTORY (#PCDATA)>
<!ELEMENT CHROOT_DIRECTORY (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>

<!ELEMENT CREATED (DATETIME, BY)> 
<!ELEMENT BY (#PCDATA)>
<!ELEMENT LAST_MODIFIED (DATETIME)>
<!ELEMENT COMMENTS (#PCDATA)>

<!ELEMENT WARNING_LIST (WARNING+)>
<!ELEMENT WARNING (CODE?, TEXT, URL?, ID_SET?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT ID_SET (ID|ID_RANGE)+>
<!ELEMENT ID_RANGE (#PCDATA)>

<!ELEMENT GLOSSARY (USER_LIST?)>
<!ELEMENT USER_LIST (USER+)> 
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>

<!-- EOF -->

**DTDs for auth type "dns_bind"**

*<platform API server>/api/2.0/batch_return.dtd>*

*<platform API server>/api/2.0/fo/auth/auth_bind_list_output.dtd*>
Docker Record

/api/2.0/fo/auth/docker/

[POST]

Create, update, list and delete Docker records for compliance scans (using PC). This record is used to authenticate to a Docker daemon (version 1.9 to 1.12) running on a Linux host.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) The record title.</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) User defined comments.</td>
</tr>
<tr>
<td>docker_deamon_conf_file={value}</td>
<td>(Optional to create or update record) Location of the configuration file for the docker daemon.</td>
</tr>
<tr>
<td>docker_command={value}</td>
<td>(Optional) The docker command to connect to a local docker daemon.</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required to create record) IPs to be added to your docker record.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional and valid only to update record) IPs to be added to an existing record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional and valid to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>network_id={1</td>
<td>0}</td>
</tr>
</tbody>
</table>
Sample - Create Docker record

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl demo" -d "action=create&title=docker_sample&ips=10.10.30.159&docker_deamon_conf_file=/etc/docker/daemon.json&docker_command=/usr/bin/docker&echo_request=1" "https://qualysapi.qualys.com/api/2.0/fo/auth/docker/"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
<REQUEST>
<DATETIME>2018-03-09T06:09:46Z</DATETIME>
<USER_LOGIN>username</USER_LOGIN>
<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/auth/docker/</RESOURCE>
<PARAM_LIST>
<PARAM>
(KEY)action</KEY>
<VALUE>create</VALUE>
</PARAM>
<PARAM>
(KEY)title</KEY>
<VALUE>docker_sample</VALUE>
</PARAM>
<PARAM>
(KEY)ips</KEY>
<VALUE>10.10.30.159</VALUE>
</PARAM>
<PARAM>
(KEY)docker_deamon_conf_file</KEY>
<VALUE>/etc/docker/daemon.json</VALUE>
</PARAM>
<PARAM>
(KEY)docker_command</KEY>
<VALUE>/usr/bin/docker</VALUE>
</PARAM>
<PARAM>
(KEY)echo_request</KEY>
<VALUE>1</VALUE>
</PARAM>
</PARAM_LIST>
</REQUEST>
```
Sample - Update Docker Record

**API request:**

```
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-03-09T06:12:57Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>72685</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
    </RESPONSE>
  </BATCH_RETURN>
```

**DTDs for auth type “docker”**

- `<platform API server>/api/2.0/batch_return.dtd`
- `<platform API server>/api/2.0/fo/auth/docker/auth_docker_list_output.dtd`
HTTP Record

/.api/2.0/fo/auth/http/

[POST]

Create, update and delete HTTP records for authenticated scans of protected portions of web sites and devices, like printers and routers, that require HTTP protocol level authentication. Vulnerability scans are supported (using VM).

How it works - During a vulnerability scan, if we come across a web page that requires HTTP authentication then we’ll check to see if an HTTP record exists in your account with applicable credentials. If yes, we’ll use the credentials in the record to perform HTTP authentication. (Note this is not Form-based authentication.)

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={value}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional for create or update request) User-defined comments.</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) One or more HTTP record IDs.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required for a create request; Optional for an update request; otherwise invalid) The HTTP record title.</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The user name to be used for authentication.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required to create record, optional to update record) The password to be used for authentication. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td>vhost={value} - or - realm={value}</td>
<td>(Required to create record; optional to update record) Specify the protected device or web page you want to authenticate against. You can specify a virtual host (an FQDN such as vhost=bank.qualys.com) or the name of a realm (realm=My+Homepage).</td>
</tr>
<tr>
<td>ssl={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
Sample - Create HTTP record, realm

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=create&username=jsmith&password=<PASSWORD>&title=My+HTTP+Record+1&realm=My+Homepage" "https://qualysapi.qualys.com/api/2.0/fo/auth/http/
```

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-01-03T07:51:48Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>55111</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Create HTTP record, virtual host

API request:
```bash
```

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-01-03T08:02:44Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>55112</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
DTDs for auth type “http”

<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/http/auth_http_list_output.dtd
IBM DB2 Record

/api/2.0/fo/auth/ibm_db2/

[POST]

Create, update, list and delete IBM DB2 records for vulnerability and compliance scans (using VM, PC). This record is used for authenticated scanning of one or more DB2 instances on a single host. Want to scan multiple instances? See “Multiple DB2 Instances” in online help.

Requirement - You must set up target hosts per the Qualys User Guide.

Download Qualys User Guide - IBM DB2 Authentication (.zip)

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>vault_id={value}</td>
<td>(Required only when action=create and login_type=vault) The ID of the vault you want to use to retrieve the password for login.</td>
</tr>
<tr>
<td>vault_type={value}</td>
<td>(Required only when action=create and login_type=vault) The third party vault to be used to retrieve the password for login. Certain vaults support this capability. See “Vault Support Matrix” in the API User Guide.</td>
</tr>
</tbody>
</table>

The following vault types are supported for IBM DB2 at this time: ARCON PAM, CA Access Control, CyberArk AIM, CyberArk PIM Suite, HashiCorp, Lieberman ERPM, Quest Vault, Thycotic Secret Server

[vault parameters] | (Required only when action=create and login_type=vault) Vault specific parameters required depend on the vault type you’ve selected. See “Vault Definition” in the API User Guide to know which parameters are required for each vault type. |
| title={value} | (Required to create record) The title for the record. The title must be unique and may include a maximum of 255 characters (ascii). |
Scan Authentication
IBM DB2 Record

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>comments={value}</td>
<td>(Optional) User defined notes about the record. Maximum of 1999 characters (ascii).</td>
</tr>
<tr>
<td>pc_only={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

**Login Credentials**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>login_type={basic</td>
<td>vault}</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The user name for a DB2 database account. A maximum of 13 characters (ascii) may be specified.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required to create record, optional to update record) The password for a DB2 database account. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td>database={value}</td>
<td>(Required to create record, optional to update record) The name of the DB2 database. A maximum of 8 characters (ascii) may be specified.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Required to create record, optional to update record) The port the database instance is running on.</td>
</tr>
</tbody>
</table>

**Target Hosts**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record, optional to update record) Add IP addresses of the hosts you want to scan using this record. Overwrites (replaces) the IP address(es) in the IP list for an existing authentication record. The IPs you specify are added, and any existing IPs are removed. You may enter a combination of IPs and IP ranges.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IP address(es) to the IP list for an existing authentication record. You may enter a combination of IPs and IP ranges.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional and valid to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>

**OS Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>win_db2dir={value}</td>
<td>The path to the DB2 runtime library if you want the service to perform OS-dependent compliance checks. This is the location where DB2 has been installed on the server. Maximum of 255 characters.</td>
</tr>
<tr>
<td>unix_db2dir={value}</td>
<td></td>
</tr>
</tbody>
</table>
Sample - Create IBM DB2 Record with Vault

In this sample, we’re creating a new record and specifying a CyberArk AIM vault.

API request:
```
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>

Parameter Description
---
win_prilogfile={value} The path to the primary archive location if you want the service to perform OS-dependent compliance checks. This is the directory where the primary log files are located. Maximum of 255 characters.
unix_prilogfile={value}

win_seclogfile={value} The path to the secondary archive location if you want the service to perform OS-dependent compliance checks. Maximum of 255 characters. This parameter specifies the number of secondary log files that are created and used for recovery log files (only as needed). It is set by the DB2 logsecond parameter.
unix_seclogfile={value}

win_terlogfile={value} The path to the tertiary archive location if you want the service to perform OS-dependent compliance checks. Maximum 255 characters.
unix_terlogfile={value} This parameter specifies a path to which DB2 will try to archive log files if the log files cannot be archived to either the primary or the secondary (if set) archivestopaths because of a media problem affecting those destinations. It is set by the DB2 failarchpath parameter.

win_mirlogfile={value} The path to the mirror archive location if you want the service to perform OS-dependent compliance checks. Maximum 255 characters.
unix_mirlogfile={value} If mirrorlogpath is configured, DB2 will create active log files in both the log path and the mirror log path. All log data will be written to both paths. The mirror log path has a duplicate set of active log files. If the active log files are destroyed by a disk error or human error, the database can still function.
Multiple DB2 Instances

The service has the ability to authenticate to multiple DB2 instances on a single host during scanning. For a vulnerability scan, an instance “uniqueness” is defined by an IP address and port. For a compliance scan, an instance “uniqueness” is defined by an IP address, port and database name. The setting for “pc_only” has an impact on how the services determines the uniqueness of a DB2 instance.

Let’s say you want to define these DB2 records in your account.

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Port</th>
<th>Database Name</th>
<th>pc_only=0/1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record 1</td>
<td>10.10.31.178</td>
<td>5000</td>
<td>SAMPLE</td>
</tr>
<tr>
<td>Record 2</td>
<td>10.10.30.159</td>
<td>5000</td>
<td>TOOLS</td>
</tr>
<tr>
<td>Record 3</td>
<td>10.10.30.159</td>
<td>5000</td>
<td>SAMPLE</td>
</tr>
</tbody>
</table>

Record 1 and Record 2 will be used for vulnerability scans and compliance scans. You’ll notice Records 2 and 3 have the same IP address and port but different database names - this is allowed because Record 3 is used for compliance scans only.
DB2 Paths
When specifying the path to configuration files, these special characters are not allowed:
For Windows:
; & | # % ? ! ` * ( ) [ ] " ' > = ^ /
For Unix:
; & | # % ? ! ` * ( ) [ ] " ' > = ^ \n
DTDs for auth type “ibm_db2”
<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/ibm_db2/auth_ibm_db2_list_output.dtd
InformixDB Record

/api/2.0/fo/auth/informixdb/

[POST]

Create, update, list and delete InformixDB authentication records. Compliance scans are supported (using PC).

- Unix authentication is required for compliance scans using the PC app. Make sure the IP addresses you define in your InformixDB records are also defined in Unix records.

- We strongly recommend you create one or more dedicated user accounts to be used solely by the Qualys Cloud Platform to authenticate to InformixDB instances.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
<tr>
<td></td>
<td>- If ssl_verify=0, the Qualys scanners authenticate with Informix Servers that don’t use SSL or InformixDB servers that use SSL. However, in the SSL case, the server SSL certificate verification will be skipped.</td>
</tr>
<tr>
<td></td>
<td>- If unspecified (or ssl_verify=1), the Qualys scanners will only send a login request after verifying that a connection to the InformixDB server uses SSL, the server SSL certificate is valid and matches the scanned host.</td>
</tr>
<tr>
<td>hosts={value}</td>
<td>(Optional to create or update record) A list of FQDNs for the hosts that correspond to all host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed. Multiple hosts are comma separated.</td>
</tr>
<tr>
<td>database_name={value}</td>
<td>(Required to create record, optional to update record) The database name to authenticate to. Specify a valid InformixDB database name. Maximum 255 characters.</td>
</tr>
</tbody>
</table>
**Sample - Create InformixDB record (with basic login and without ssl_verify)**

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=create&title=my-informixdb-record&username=informix-admin&password=<PASSWORD>&ips=10.10.10.11&comments=informix-basic-ipv4&unix_config_dir=/opt/informix&port=1526&ssl_verify=0&unix_on_config_dir=/opt/Informix/etc/onconfig.demo&unix_sql_host_dir=/opt/Informix/etc/sqlhosts.demo&database_name=dbname&login_type=basic"
```
"https://qualysapi.qualys.com/api/2.0/fo/auth/informixdb/"

**XML output:**

```xml
<RESPONSE>
    <DATETIME>2019-01-30T15:45:05Z</DATETIME>
    <BATCH_LIST>
        <BATCH>
            <TEXT>Successfully Created</TEXT>
            <ID_SET>
                <ID>43025</ID>
            </ID_SET>
        </BATCH>
    </BATCH_LIST>
</RESPONSE>
</BATCH_RETURN>
```

**Sample - Create InformixDB record (with ssl_verify)**

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=create&title=my-informixdb-record&username=informix-admin&password=informix-basect unix_config_dir=/opt/informix&port=1526&ssl_verify=1&un
ix_on_config_dir=/opt/Informix/etc/onconfig.demo&unix_sql_host_dir=/opt/Informix/etc/sqlhosts.demo&database_name=dbname&login_type=basic&hosts=mlinformixdb32e.s2012r2.qualys.com,mlinformixdb32e.s2008r2.qualys.com" "https://qualysapi.qualys.com/api/2.0/fo/auth/informixdb/
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"http://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
    <RESPONSE>
        <DATETIME>2019-01-30T15:47:01Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Created</TEXT>
                <ID_SET>
                    <ID>43026</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```
Scan Authentication
InformixDB Record

Sample - List InformixDB record

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list&details=Basic"
   "https://qualysapi.qualys.com/api/2.0/fo/auth/informixdb/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_INFORMIXDB_LIST_OUTPUT SYSTEM
   "https://qualysapi.qualys.com/fo/auth/informixdb/auth_informixdb_list_output.dtd">
<AUTH_INFORMIXDB_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2019-01-30T15:19:02Z</DATETIME>
    <AUTH_INFORMIXDB_LIST>
      <AUTH_INFORMIXDB>
        <ID>40034</ID>
        <TITLE><![CDATA[InformixDB1]]></TITLE>
        <USERNAME><![CDATA[root]]></USERNAME>
        <DATABASE><![CDATA[informixdb]]></DATABASE>
        ...
      </AUTH_INFORMIXDB>
    </AUTH_INFORMIXDB_LIST>
  </RESPONSE>
</AUTH_INFORMIXDB_LIST_OUTPUT>
```

Sample - Update InformixDB record

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d
   "action=update&ids=41026&title=API-informixdb-basic-login-updated&username=admin-updated-again&password=<updated-password>&database_name=new-admin&comments=informixdb-basic-login-ipv4-updated&unix_config_dir=/opt/informixdb/updated/again"
   "https://qualysapi.qualys.com/api/2.0/fo/auth/informixdb/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
   "https://qualysapi.qualys.com/fo/auth/informixdb/batch_return.dtd">
</BATCH_RETURN>
```
Sample - Delete InformixDB record

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=delete&ids=43023,43024"
"https://qualysapi.qualys.com/api/2.0/fo/auth/informixdb/"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"http://10.114.69.159:46445/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-01-30T15:41:46Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID_RANGE>43023-43024</ID_RANGE>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

DTDs for auth type “informixdb”

<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/informixdb/auth_informixdb_list_output.dtd
Infoblox Record

/api/2.0/fo/auth/infoblox

[POST]

Create, update, list and delete Infoblox authentication records. Infoblox authentication records are available for the PC/SCA module.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to edit or delete record) Record IDs to edit/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) for the Infoblox devices you want to authenticate to. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td>api_version=value</td>
<td>(Required) API version required on Infoblox devices.</td>
</tr>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

Login credentials

| login_type={basic|vault} | (Optional) The login type is basic by default. You can choose vault (for vault based authentication). |
| username={value}         | (Required to create record, optional to update record) The username to be used for authentication to Infoblox server. |
| password={value}         | (Required to create record, optional to update record) The password to be used for authentication to Infoblox server. |

Vault

| vault_type={value} | (Required to create record when login_type=vault) The vault type to be used for authentication. See Vault Support matrix. |
Scan Authentication
Infoblox Record

Sample - Create Infoblox record (with basic login)

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=create&ips=10.20.30.40&title=Infoblox_Auth11&api_version=1
v2.0123&ssl_verify=false&username=joe_user&password=<PASSWORD>" 
"https://qualysapi.qualys.com/api/2.0/fo/auth/infoblox/" > file.xml
```

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2022-06-29T10:50:19Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>6317683</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - List Infoblox record

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d "action=list" 
"https://qualysapi.qualys.com/api/2.0/fo/auth/infoblox/" > file.xml
```

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_INFOBLOX_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/infoblox/auth_infoblox.dtd">
```

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vault_id=value</td>
<td>(Required to create record when login_type=vault and you want to retrieve private key from vault) The vault ID where you want to retrieve the private key from. Certain vaults support this capability.</td>
</tr>
<tr>
<td>[vault parameters]</td>
<td>(Required to create record when login_type=vault) Vault specific parameters required depend on the vault type you’ve selected. See Vault Definition.</td>
</tr>
</tbody>
</table>
Sample - Update Infoblox record (with basic login)

API request:


XML Output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2022-06-29T11:10:12Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
```

297
<TEXT>Successfully Updated</TEXT>
<ID_SET>
  <ID>6317683</ID>
</ID_SET>
</BATCH>
</BATCH_LIST>
</RESPONSE>
</BATCH_RETURN>

Sample - Delete Infoblox record

**API request:**

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=delete&ids=6317685"
"https://qualysapi.qualys.com/api/2.0/fo/auth/infoblox/" > file.xml
```

**XML Output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2022-06-29T11:18:52Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>6317683</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Create Infoblox record (with Vault login)

**API request:**

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=create&ips=10.20.30.40&title=Infoblox_Auth_With_Vault&login_type=vault&api_version=2.0&username=joe_user&vault_type=Has
corp&vault_id=1062779&secret_kv_name=admin&secret_kv_key=Infoblox_vault_secret&ssl_verify=0"
"https://qualysapi.qualys.com/api/2.0/fo/auth/infoblox/" > file.xml
```

**XML Output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2022-06-29T11:18:52Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>6317683</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2022-07-14T07:46:29Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>1898844</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

Sample - Update Infoblox record (with Vault login)

API request:


XML Output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2022-07-14T07:46:29Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>1898844</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
DTDs for auth type "Infoblox"

<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/infoblox/auth_infoblox_list_output.dtd
**JBoss Server record**

/api/2.0/fo/auth/jboss/

[POST]

Create, update, list and delete JBoss Server records for vulnerability and compliance scans (using VM, PC). Supports Windows and Unix platforms.

Supported technologies:

Windows - WildFly/JBoss EAP

Unix - WildFly/JBoss EAP

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required) Specify a single or comma separated valid JBoss type auth record ID(s).</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique.</td>
</tr>
<tr>
<td>comment={value}</td>
<td>(Optional to create or update record) User defined comments.</td>
</tr>
</tbody>
</table>

**Windows platform**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>windows_working_mode={value}</td>
<td>(Optional) Input values should be standalone_mode or domain_controller_mode.</td>
</tr>
<tr>
<td>windows_home_path={value}</td>
<td>Required if windows working mode is selected.</td>
</tr>
<tr>
<td>windows_base_path={value}</td>
<td>Required if windows working mode is selected.</td>
</tr>
<tr>
<td>windows_conf_dir_path={value}</td>
<td>Required if windows working mode is selected.</td>
</tr>
<tr>
<td>windows_conf_file_path={value}</td>
<td>Required if windows working mode is selected.</td>
</tr>
<tr>
<td>windows_conf_host_file_path={value}</td>
<td>Required if selected Windows working mode is domain controller.</td>
</tr>
</tbody>
</table>

**Unix platform**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unix_working_mode={value}</td>
<td>(Optional) Input values should be standalone_mode or domain_controller_mode.</td>
</tr>
<tr>
<td>unix_home_path={value}</td>
<td>Required if Unix working mode is selected.</td>
</tr>
</tbody>
</table>
Scan Authentication
JBoss Server record

Sample - Create JBoss Server record

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"action=create&title=jbos_rec&windows_working_mode=standalone_mode
&windows_base_path=c:\&windows_home_path=c:\&windows_conf_file_path=c:\&windows_conf_dir_path=c:\&comment=record
creation&ips=10.10.10.224"
"https://qualysapi.qualys.com/api/2.0/fo/auth/jboss/
```

XML output:
```xml
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-08-03T10:42:32Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>296004</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unix_base_path= {value}</td>
<td>Required if Unix working mode is selected.</td>
</tr>
<tr>
<td>unix_conf_dir_path= {value}</td>
<td>Required if Unix working mode is selected.</td>
</tr>
<tr>
<td>unix_conf_file_path= {value}</td>
<td>Required if Unix working mode is selected.</td>
</tr>
<tr>
<td>unix_conf_host_file_path= {value}</td>
<td>Required if selected Unix working mode is domain controller.</td>
</tr>
<tr>
<td>ips= {value}</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td>add_ips= {value}</td>
<td>(Optional and valid only to update record) IPs to be added to an existing record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>remove_ips= {value}</td>
<td>(Optional and valid to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>network_id= {value}</td>
<td>(Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>

Target Hosts

ips= {value} (Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.

add_ips= {value} (Optional and valid only to update record) IPs to be added to an existing record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.

remove_ips= {value} (Optional and valid to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.

network_id= {value} (Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record.
Sample - List JBoss Server record

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -d "action=list&ids=296004"
"https://qualysapi.qualys.com/api/2.0/fo/auth/jboss/

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_JBOSS_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/jboss/auth_jboss_list_output.dtd">
<AUTH_JBOSS_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-08-03T10:44:39Z</DATETIME>
    <AUTH_JBOSS_LIST>
      <AUTH_JBOSS>
        <ID>296004</ID>
        <TITLE><![CDATA[jboss_record]]></TITLE>
        <IP_SET>
          <IP>10.10.10.224</IP>
        </IP_SET>
        <WINDOWS>
          <HOME_PATH><![CDATA[c:\]]></HOME_PATH>
          <DOMAIN_MODE><![CDATA[true]]></DOMAIN_MODE>
          <BASE_PATH><![CDATA[c:\]]></BASE_PATH>
          <CONF_DIR_PATH><![CDATA[c:\]]></CONF_DIR_PATH>
          <CONF_FILE_PATH><![CDATA[c:\]]></CONF_FILE_PATH>
          <CONF_HOST_FILE_PATH><![CDATA[c:\]]></CONF_HOST_FILE_PATH>
        </WINDOWS>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2018-08-03T10:42:32Z</DATETIME>
          <BY>abc_pk</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2018-08-03T10:43:58Z</DATETIME>
        </LAST_MODIFIED>
        <COMMENTS><![CDATA[record creation]]></COMMENTS>
      </AUTH_JBOSS>
    </AUTH_JBOSS_LIST>
  </RESPONSE>
</AUTH_JBOSS_LIST_OUTPUT>
Sample record configurations
We have sample JBoss record configurations in our online help. Log in to your Qualys account and select Help > Online Help and search for JBoss.

DTDs for auth type “jboss”
<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/jboss/auth_jboss_list_output.dtd
# Kubernetes Record

/api/2.0/fo/auth/kubernetes/

[POST]

Create, update, list and delete Kubernetes records for compliance scans (using PC). This record is used to authenticate to a Kubernetes application (version 1.x) running on a Unix host.

## Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>details={Basic}</td>
<td>(Optional) Default value is Basic. You can choose from None, Basic, and All.</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to list, update or delete record and optional to create record) Kubernetes authentication record IDs. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma-separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) The title of the record. The title must be unique and may include a maximum of 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) User-defined notes about the record. Maximum of 1999 characters (ascii).</td>
</tr>
<tr>
<td>unix_bin_path={value}</td>
<td>(Optional) Absolute path of the 'kubectl' command.</td>
</tr>
<tr>
<td>unix_conf_path={value}</td>
<td>(Optional) Absolute path of the Kubernetes configuration file.</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP addresses for the Kubernetes targets you want to authenticate to. Multiple entries are comma-separated. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional and valid only to update record) IPs to be added to an existing record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.</td>
</tr>
</tbody>
</table>
Sample - Create Kubernetes record

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=create&title=kubernetes auth record&unix_bin_path=/usr/bin/kubectl&unix_conf_path=/root/kube/config&ips=10.10.10.10&comments=kube auth record" "https://qualysapi.qualys.com/api/2.0/fo/auth/kubernetes/
```

**XML output:**
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-08-30T11:30:58Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>94170</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Update Kubernetes record

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=update&ids=10001&title=kubernetes auth record&unix_bin_path=/usr/bin/kubectl&unix_conf_path=/root/kube/config" "https://qualysapi.qualys.com/api/2.0/fo/auth/kubernetes/
```

**XML output:**
```xml
<remove_ips={value} (Optional and valid to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network_id={value}</td>
<td>(Optional, and valid when the Networks feature is enabled) The network ID for the record. By default, the parameter is set to 0.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>remove_ips=(value)</td>
<td>(Optional and valid to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
</tbody>
</table>

Sample - Update Kubernetes record

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=update&ids=10001&title=kubernetes auth record&unix_bin_path=/usr/bin/kubectl&unix_conf_path=/root/kube/config" "https://qualysapi.qualys.com/api/2.0/fo/auth/kubernetes/
```

**XML output:**
```xml
```
<xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-08-30T12:30:58Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>94170</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

DTDs for auth type “kubernetes”
<platform API server>/api/2.0/fo/auth/auth_records.dtd
<platform API server>/api/2.0/fo/auth/kubernetes/auth_kubernetes_list_output.dtd
MariaDB Record

/api/2.0/fo/auth/mariadb/

[POST]

Create, update, list and delete MariaDB authentication records. Compliance scans are supported (using PC).

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
<tr>
<td>hosts={value}</td>
<td>(Optional to create or update record) A list of FQDNs for the hosts that correspond to all host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed. Multiple hosts are comma separated.</td>
</tr>
<tr>
<td>database={value}</td>
<td>(Required to create record, optional to update record) The database name to authenticate to. Specify a valid MariaDB database name.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Required to create record, optional to update record) The port the database name is running on. The default is 3306.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>windows_config_file={value}</td>
<td>(Optional to create or update record) The path to the Windows mariadb config file. Access to this config file is required to run certain checks on Windows hosts. Note: You must include one or both of these parameters in a create request: windows_config_file and unix_config_file.</td>
</tr>
<tr>
<td>unix_config_file={value}</td>
<td>(Optional to create or update record) The path to the Unix mariadb config file. Access to this config file is required to run certain checks on Unix hosts. Note: You must include one or both of these parameters in a create request: windows_config_file and unix_config_file.</td>
</tr>
<tr>
<td>client_cert={value}</td>
<td>(Optional to create or update record) PEM-encoded X.509 certificate. Specify if certificate authentication is required by your server to establish an SSL connection.</td>
</tr>
<tr>
<td>client_key={value}</td>
<td>(Optional to create or update record) PEM-encoded RSA private key. Specify if certificate authentication is required by your server to establish an SSL connection.</td>
</tr>
</tbody>
</table>

### Login credentials

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>login_type={basic</td>
<td>vault}</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The username to be used for authentication to MariaDB server.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required to create record, optional to update record) The password to be used for authentication to MariaDB server.</td>
</tr>
</tbody>
</table>

### Vault

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vault_type={value}</td>
<td>(Required to create record when login_type=vault) The vault type to be used for authentication. See Vault Support matrix.</td>
</tr>
<tr>
<td>vault_id={value}</td>
<td>(Required to create record when login_type=vault and you want to retrieve private key from vault) The vault ID where you want to retrieve the private key from. Certain vaults support this capability.</td>
</tr>
<tr>
<td>{vault parameters}</td>
<td>(Required to create record when login_type=vault) Vault specific parameters required depend on the vault type you’ve selected. See Vault Definition.</td>
</tr>
</tbody>
</table>

### Target Hosts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs to the IPs list for this record. Multiple IPs/ranges are comma separated.</td>
</tr>
</tbody>
</table>
Sample - Create MariaDB record (with basic login)

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=create&title=MariaDB_Auth1&username=root&password=<PASSWORD>&ips=10.10.31.86&echo_request=0&unix_config_file=/etc/my.cnf&port=22&database=mariadb" "https://qualysapi.qualys.com/api/2.0/fo/auth/mariadb/
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-07-17T21:56:47Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>284866</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

**Sample - List MariaDB records**

Use the new MariaDB Authentication Record List API (/api/2.0/fo/auth/mariadb/?action=list) to list MariaDB records.

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list" "https://qualysapi.qualys.com/api/2.0/fo/auth/mariadb/
```
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_MARIADB_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/mariadb/auth_mariadb_list_output.dtd">

<AUTH_MARIADB_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-07-17T21:57:32Z</DATETIME>
    <AUTH_MARIADB_LIST>
      <AUTH_MARIADB>
        <ID>284866</ID>
        <TITLE><![CDATA[MariaDB_Auth1]]></TITLE>
        <USERNAME><![CDATA[root]]></USERNAME>
        <DATABASE><![CDATA[mariadb]]></DATABASE>
        <PORT>22</PORT>
        <IP_SET>
          <IP>10.10.31.86</IP>
        </IP_SET>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <SSL_VERIFY>false</SSL_VERIFY>
        <WINDOWS_CONF_FILE><![CDATA[]]></WINDOWS_CONF_FILE>
        <UNIX_CONF_FILE><![CDATA[/etc/my.cnf]]></UNIX_CONF_FILE>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2018-07-17T21:56:47Z</DATETIME>
          <BY>seenu_yn</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2018-07-17T21:56:47Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_MARIADB>
    </AUTH_MARIADB_LIST>
  </RESPONSE>
</AUTH_MARIADB_LIST_OUTPUT>
```

DTDs for auth type “mariadb”

<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/mariadb/auth_mariadb_list_output.dtd
Microsoft SharePoint Record

/api/2.0/fo/auth/microsoft_sharepoint/

[POST]

List, create, update, and delete Microsoft SharePoint records for authenticated scans of Microsoft SharePoint instances running on Windows and Database. Microsoft SharePoint version 2010, 2013, 2016, and 2019 are supported.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>db_local={0</td>
<td>1}</td>
</tr>
<tr>
<td>windows_domain={value}</td>
<td>(Required when db_local=0, otherwise invalid) The domain name where the login credentials are stored when the login credentials are for a Microsoft Windows operating system account that is associated with a MS SQL Server database account. The domain name may include 1-256 characters (ascii). For an update request when the credentials for the record are for a Microsoft Windows account (db_local=0) and you want to change the record to use credentials for a MS SQL Server account (db_local=1), then you must set windows_domain=&quot;&quot; (the empty string) to clear the current parameter setting.</td>
</tr>
<tr>
<td>kerberos={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
### Scan Authentication

Microsoft SharePoint Record

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntlmv2={0</td>
<td>1}</td>
</tr>
<tr>
<td>ntlmv1={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

### Login credentials

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username={value}</td>
<td>(Required for create request) The username of the account to be used for authentication. If password is specified this is the username of a MS SQL Server database user account used for SharePoint. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>password={value}</td>
<td>(For create request, password or login_type=vault is required) The password of the MS SQL Server database user account to be used for authentication. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td>login_type={value}</td>
<td>(For create request, password or login_type=vault is required) Login type can be basic (default) or vault. Set to vault if a third party vault will be used to retrieve the password. Vault parameters need to be provided in the record. See Vault Definition.</td>
</tr>
<tr>
<td>vault_id={value}</td>
<td>(Required if login_type=vault) The ID of the vault to be used to retrieve the password for login.</td>
</tr>
<tr>
<td>vault_type={value}</td>
<td>(Required if login_type=vault) The third party vault to be used to retrieve the password for login. Certain vaults support this capability. See Vault Support matrix.</td>
</tr>
<tr>
<td>secret_name={value}</td>
<td>(Required if vault type is Thycotic Secret Server) Specify the secret name that contains the password to be used for authentication. The scanning engine will perform a search for the secret name and then get the password from the secret returned by the search. A single exact match of the secret name must be found in order for authentication to be successful. The secret name may contain a maximum of 256 characters, and must not contain multibyte characters.</td>
</tr>
<tr>
<td>system_name={value}</td>
<td>(Optional if vault type is BeyondTrust PBPS or Quest Vault) The managed system name (also known as asset name). When not specified, we’ll attempt to auto-discover the system name at scan time.</td>
</tr>
<tr>
<td>account_name={value}</td>
<td>(Optional if vault type is BeyondTrust PBPS) The account name. When not specified, we’ll try the username specified in the authentication record.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>folder={value}</td>
<td>(Required if vault type is CyberArk AIM and Cyber-ARK PIM Suite) Specify the name of the folder in the secure digital safe where the password to be used for authentication should be stored. The folder name can contain a maximum of 169 characters. Entering a trailing /, as in folder/, is optional (when specified, the service removes the trailing / and does not save it in the folder name). The maximum length of a folder name with a file name is 170 characters (the leading and/or trailing space in the input value will be removed). These special characters cannot be included in a folder name: / : * ? &quot; &lt; &gt;</td>
</tr>
<tr>
<td>file={value}</td>
<td>(Required if vault type is CyberArk AIM and Cyber-ARK PIM Suite) Specify the name of the file in the secure digital safe where the password to be used for authentication should be stored. The file name can contain a maximum of 165 characters. The maximum length of a folder name plus a file name is 170 characters (the leading and/or trailing space in the input value will be removed). These special characters cannot be included in a file name: / : * ? &quot; &lt; &gt;</td>
</tr>
</tbody>
</table>

### Target Hosts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) for the Microsoft SharePoint targets you want to authenticate to. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional to create or update record, and valid only when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>
Sample: List all Records

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list" "https://qualysapi.qualys.com/api/2.0/fo/auth/"

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_RECORDS_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/auth_records.dtd">
<AUTH_RECORDS_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-02-14T06:40:29Z</DATETIME>
    <AUTH_RECORDS>
      <AUTH_UNIX_IDS>
        <ID_SET>
          <ID>63215</ID>
          <ID>63239</ID>
          <ID>65170</ID>
          <ID>65172</ID>
          <ID>66185</ID>
        </ID_SET>
      </AUTH_UNIX_IDS>
      <AUTH_VMWARE_IDS>
        <ID_SET>
          <ID>63213</ID>
          <ID>63235</ID>
          <ID>63237</ID>
          <ID>63241</ID>
        </ID_SET>
      </AUTH_VMWARE_IDS>
      <AUTH_POSTGRESQL_IDS>
        <ID_SET>
          <ID>66387</ID>
          <ID>66389</ID>
          <ID>69602</ID>
          <ID>72224</ID>
        </ID_SET>
      </AUTH_POSTGRESQL_IDS>
      <AUTH_ORACLE_HTTP_SERVER_IDS>
        <ID_SET>
          <ID>66388</ID>
        </ID_SET>
      </AUTH_ORACLE_HTTP_SERVER_IDS>
      <AUTH_MICROSOFT_SHAREPOINT_IDS>
        <ID_SET>
```
Sample - List Microsoft SharePoint Records with Basic Details

API request:
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: Curl' -d "action=list&details=Basic" "https://qualysapi.qualys.com/api/2.0/fo/auth/microsoft_sharepoint/"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_MICROSOFT_SHAREPOINT_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/microsoft_sharepoint/auth_microsoft_sharepoint_list_output.dtd">
<AUTH_MICROSOFT_SHAREPOINT_LIST>
  <AUTH_MICROSOFT_SHAREPOINT>
    <ID>2372474</ID>
    <TITLE><![CDATA[SharePoint_WindowsAuth]]></TITLE>
    <USERNAME><![CDATA[username]]></USERNAME>
    <IP_SET>
      <IP>10.10.10.13</IP>
    </IP_SET>
    <MSSQL>
      <DB_LOCAL><![CDATA[0]]></DB_LOCAL>
      <WINDOWS_DOMAIN><![CDATA[sample.qualys.com]]></WINDOWS_DOMAIN>
      <KERBEROS><![CDATA[1]]></KERBEROS>
      <NTLMV2><![CDATA[1]]></NTLMV2>
    </MSSQL>
    <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
    <CREATED>
      <DATETIME>2020-03-10T18:47:26Z</DATETIME>
      <BY>joe_user</BY>
    </CREATED>
    <LAST_MODIFIED>
      <DATETIME>2020-03-10T18:47:26Z</DATETIME>
    </LAST_MODIFIED>
  </AUTH_MICROSOFT_SHAREPOINT>
  <AUTH_MICROSOFT_SHAREPOINT>
    <ID>2372483</ID>
  </AUTH_MICROSOFT_SHAREPOINT>
</AUTH_MICROSOFT_SHAREPOINT_LIST>
<TITLE><![CDATA[SharePoint_DatabaseAuth]]></TITLE>
<USERNAME><![CDATA[username]]></USERNAME>
<IP_SET>
  <IP_RANGE>10.10.10.19-10.10.10.20</IP_RANGE>
</IP_SET>
<MSSQL>
  <DB_LOCAL><![CDATA[1]]></DB_LOCAL>
  <KERBEROS><![CDATA[1]]></KERBEROS>
  <NTLMV2><![CDATA[1]]></NTLMV2>
  <NTLMV1><![CDATA[1]]></NTLMV1>
</MSSQL>
<LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
<CREATED>
  <DATETIME>2020-03-10T20:53:37Z</DATETIME>
  <BY>joe_user</BY>
</CREATED>
<LAST_MODIFIED>
  <DATETIME>2020-03-10T20:53:37Z</DATETIME>
</LAST_MODIFIED>
</AUTH_MICROSOFT_SHAREPOINT>

Sample - List Microsoft SharePoint Records with All Details

**API request:**
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: Curl' -d "action=list&details=All" "https://qualysapi.qualys.com/api/2.0/fo/auth/microsoft_sharepoint/

**XML output:**
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_MICROSOFT_SHAREPOINT_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/microsoft_sharepoint/auth_microsoft_sharepoint_list_output.dtd">
<AUTH_MICROSOFT_SHAREPOINT_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-03-11T22:56:20Z</DATETIME>
    <AUTH_MICROSOFT_SHAREPOINT_LIST>
      <AUTH_MICROSOFT_SHAREPOINT>
        <ID>2372474</ID>
        <TITLE><![CDATA[SharePoint_WindowsAuth]]></TITLE>
        <USERNAME><![CDATA[username]]></USERNAME>
        <IP_SET>
          <IP>10.10.10.13</IP>
        </IP_SET>
        <MSSQL>
          <DB_LOCAL><![CDATA[1]]></DB_LOCAL>
          <KERBEROS><![CDATA[1]]></KERBEROS>
          <NTLMV2><![CDATA[1]]></NTLMV2>
          <NTLMV1><![CDATA[1]]></NTLMV1>
        </MSSQL>
      </AUTH_MICROSOFT_SHAREPOINT>
    </AUTH_MICROSOFT_SHAREPOINT_LIST>
  </RESPONSE>
</AUTH_MICROSOFT_SHAREPOINT_LIST_OUTPUT>
<DB_LOCAL><![CDATA[0]]></DB_LOCAL>

<WINDOWS_DOMAIN><![CDATA[sample.qualys.com]]></WINDOWS_DOMAIN>
  <KERBEROS><![CDATA[1]]></KERBEROS>
  <NTLMV2><![CDATA[1]]></NTLMV2>
</MSSQL>
<LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
<CREATED>
  <DATETIME>2020-03-10T18:47:26Z</DATETIME>
  <BY>joe_user</BY>
</CREATED>
<LAST_MODIFIED>
  <DATETIME>2020-03-10T18:47:26Z</DATETIME>
</LAST_MODIFIED>
</AUTH_MICROSOFT_SHAREPOINT>

<AUTH_MICROSOFT_SHAREPOINT>
  <ID>2372483</ID>
  <TITLE><![CDATA[SharePoint_DatabaseAuth]]></TITLE>
  <USERNAME><![CDATA[username]]></USERNAME>
  <IP_SET>
    <IP_RANGE>10.10.10.19-10.10.10.20</IP_RANGE>
  </IP_SET>
  <MSSQL>
    <DB_LOCAL><![CDATA[1]]></DB_LOCAL>
    <KERBEROS><![CDATA[1]]></KERBEROS>
    <NTLMV2><![CDATA[1]]></NTLMV2>
    <NTLMV1><![CDATA[1]]></NTLMV1>
  </MSSQL>
  <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
  <CREATED>
    <DATETIME>2020-03-10T20:53:37Z</DATETIME>
    <BY>joe_user</BY>
  </CREATED>
  <LAST_MODIFIED>
    <DATETIME>2020-03-10T20:53:37Z</DATETIME>
  </LAST_MODIFIED>
</AUTH_MICROSOFT_SHAREPOINT>

<AUTH_MICROSOFT_SHAREPOINT>
  <ID>2372484</ID>
  <TITLE><![CDATA[SharePoint123]]></TITLE>
  <USERNAME><![CDATA[userupdate]]></USERNAME>
  <IP_SET>
    <IP_RANGE>10.10.10.25-10.10.10.26</IP_RANGE>
  </IP_SET>
  <MSSQL>
    <DB_LOCAL><![CDATA[0]]></DB_LOCAL>
  </MSSQL>
<WINDOWS_DOMAIN><![CDATA[sample2.qualys.com]]></WINDOWS_DOMAIN>
<KERBEROS><![CDATA[1]]></KERBEROS>
<NTLMV1><![CDATA[1]]></NTLMV1>
</MSSQL>
<LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
<CREATED>
<DATETIME>2020-03-10T20:55:50Z</DATETIME>
<BY>joe_user</BY>
</CREATED>
<LAST_MODIFIED>
<DATETIME>2020-03-11T16:19:19Z</DATETIME>
</LAST_MODIFIED>
</AUTH_MICROSOFT_SHAREPOINT>
</AUTH_MICROSOFT_SHAREPOINT_LIST>
<GLOSSARY>
<USER_LIST>
<USER>
<User_LOGIN>joe_user</User_LOGIN>
<FIRST_NAME>Joe</FIRST_NAME>
<LAST_NAME>User</LAST_NAME>
</USER>
</USER_LIST>
</GLOSSARY>
</RESPONSE>
</AUTH_MICROSOFT_SHAREPOINT_LIST_OUTPUT>

Sample - Create Microsoft SharePoint Record

API request with Microsoft Windows login (db_local=0):

curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: Curl' -d
"action=create&title=SharePoint&ips=10.10.10.13&username=username&
password=<PASSWORD>&db_local=0&windows_domain=sample.qualys.com"
"https://qualysapi.qualys.com/api/2.0/fo/auth/microsoft_sharepoint/"

API request with MS SQL Server database login (db_local=1):

curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: Curl' -d
"action=create&title=SharePoint_withDatabase&ips=10.10.10.14&username=username&
password=<PASSWORD>&db_local=1"
"https://qualysapi.qualys.com/api/2.0/fo/auth/microsoft_sharepoint/"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM...>
Sample - Update Microsoft SharePoint Record

API request to update basic information:


API request to update vault login and change to different vault:

curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: Curl' -d "action=update&ids=10003&login_type=vault&vault_type=Thycotic+Secret+Server&vault_id=123&secret_name=secret-name" "https://qualysapi.qualys.com/api/2.0/fo/auth/microsoft_sharepoint/"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-02-13T07:39:09Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>72223</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Sample - Delete Microsoft SharePoint Records

API request for deleting single record:

curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: Curl' -d "action=delete&ids=10000"
"https://qualysapi.qualys.com/api/2.0/fo/auth/microsoft_sharepoint/"

API request for deleting multiple records:

curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: Curl' -d "action=list&ids=10000,10001"
"https://qualysapi.qualys.com/api/2.0/fo/auth/microsoft_sharepoint/"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-02-13T07:40:06Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>72223</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

DTDs for auth type “microsoft_sharepoint”

```
<platform API server>/api/2.0/auth_records.dtd
<platform API server>/api/2.0/fo/auth/microsoft_sharepoint/auth_microsoft_sharepoint_list_output.dtd
```
MongoDB Record

/api/2.0/fo/auth/mongodb/

[POST]

Create, update, list and delete MongoDB records for authenticated scans of MongoDB instances running on Unix. Vulnerability and compliance scans are supported (using VM, PC).

- Technologies supported: MongoDB 3.x
- For OS-level checks, make sure the IP addresses you define in your MongoDB records are also defined in Unix records.
- We strongly recommend you create one or more dedicated user accounts to be used solely by the Qualys Cloud Platform to authenticate to MongoDB instances.

Requirement - You must configure authentication credentials on target hosts.

How it works - During scanning we'll authenticate to one or more instances on a single host using all MongoDB records in your account. For compliance scans, you can scan multiple MongoDB instances on a single host and port combination.

System created authentication records supported - You can allow the system to create MongoDB authentication records for auto discovered instances and scan them. This is supported for Unix installations only. To enable this feature, you must first create MongoDB System Record Templates using the is_template input parameter and specifying login credentials. See System created MongoDB records

Download Qualys User Guide - MongoDB Authentication (.pdf)

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>is_template={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
## Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status={0</td>
<td>1}</td>
</tr>
<tr>
<td>save_as_user_auth={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={id1,id2,...}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>Target Hosts</td>
<td></td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required to create record, optional to update record) Add IP addresses of the hosts you want to scan using this record.</td>
</tr>
<tr>
<td></td>
<td>Overwrites (replaces) the IP address(es) in the IP list for an existing authentication record. The IPs you specify are added, and any existing IPs are removed. You may enter a combination of IPs and IP ranges.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IP address(es) to the IP list for an existing authentication record. You may enter a combination of IPs and IP ranges.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
<tr>
<td>MongoDB</td>
<td></td>
</tr>
<tr>
<td>unix_conf_file={value}</td>
<td>(Required for create request) The full path to the MongoDB configuration file on your Unix assets (IP addresses). The file must be in the same location on all assets for this record. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>database_name={value}</td>
<td>(Required for create request) The username of the account to be used for authentication to the database. If password is specified this is the username of a MongoDB account. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Required for create request) The port where the database instance is running. Default is 27017.</td>
</tr>
</tbody>
</table>
### Scan Authentication

**MongoDB Record**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
<tr>
<td>hosts={value}</td>
<td>(Required if ssl_verify=1) A list of FQDNs for all host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed.</td>
</tr>
</tbody>
</table>

### Login credentials

<table>
<thead>
<tr>
<th>credential_type=local</th>
<th>external</th>
<th>(Optional) The credential type is local by default which means login credential type is local authentication. You need to set credential type to external for LDAP authentication option.</th>
</tr>
</thead>
<tbody>
<tr>
<td>cleartext=0</td>
<td>1</td>
<td>(Optional) You must set credential_type to external to use cleartext parameter. The default value for cleartext is 0. You must set this parameter to 1 for successful MongoDB authentication for LDAP.</td>
</tr>
<tr>
<td>login_type={basic</td>
<td>vault</td>
<td>pkcert}</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required to create record when login_type=basic or login_type=vault) The username of the MongoDB account to be used for authentication. Maximum 100 characters (ascii).</td>
<td></td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required to create record when login_type=basic) The password of the MongoDB account to be used for authentication. Maximum 100 characters (ascii).</td>
<td></td>
</tr>
</tbody>
</table>

### Vault

<table>
<thead>
<tr>
<th>vault_type={value}</th>
<th>(Required to create record when login_type=vault) The vault type to be used for authentication. See Vault Support matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>vault_id={value}</td>
<td>(Required to create record when login_type=vault and you want to retrieve private key from vault) The vault ID where you want to retrieve the private key from. Certain vaults support this capability.</td>
</tr>
<tr>
<td>{vault parameters}</td>
<td>(Required to create record when login_type=vault) Vault specific parameters required depend on the vault type you’ve selected. See Vault Definition</td>
</tr>
<tr>
<td>private_key_vault_id={value}</td>
<td>(Required to create record when login_type=vault and you want to retrieve passphrase from vault) The vault ID where you want to retrieve the passphrase from. Certain vaults support this capability. See Vault Support matrix</td>
</tr>
<tr>
<td>passphrase_vault_id={value}</td>
<td>(For create request, required when login_type=vault and you want to retrieve passphrase from vault) The vault ID where you want to retrieve the passphrase from. Certain vaults support this capability. See Vault Support matrix</td>
</tr>
</tbody>
</table>
When we auto discover MongoDB instances, we’ll discover the target configuration for each instance but not the login credentials. We’ve introduced a new configuration called “MongoDB authentication record template” that you’ll use to provide MongoDB login credentials for system created records. You’ll create the system record template and then select it in the option profile used for discovery scans. The template is linked automatically to the system created records created as a result of the scan.

**Benefits**

- We’ll auto discover MongoDB instances on each scanned host and create authentication records for those instances. We support auto discovery and system record creation for MongoDB instances running on Unix platforms. Make sure you have Unix authentication records in your account for hosts running MongoDB.

- When we create MongoDB authentication records for discovered instances, we’ll insert the credentials from the MongoDB system record template you selected in the option profile.

- You can easily rotate MongoDB passwords. Simply edit the credentials in the MongoDB system record template and all MongoDB records linked to the template will be updated to use the new credentials with no additional scan or action by you.

- You can edit individual MongoDB system created records and save them as user created. This allows you to change the credentials for individual records without changing the credentials for all records associated with a template.

**How it works**

Here’s the basic flow for MongoDB instance discovery and auto record creation. Note - We support auto discovery and system record creation for MongoDB instances running on Unix platforms. Make sure you have Unix authentication records in your account for hosts running MongoDB.

---

### System created MongoDB records

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>private_key={value}</td>
<td><em>(For create request, required when login_type=pkcert)</em> The private key to be used for authentication. Certain vaults support this capability. See Vault Support matrix</td>
</tr>
<tr>
<td>passphrase={value}</td>
<td><em>(For create request, required when login_type=pkcert and passphrase_vault_id is not specified)</em> The private key passphrase value of an encrypted private key. Maximum 255 characters (ascii). Certain vaults support this capability. See Vault Support matrix</td>
</tr>
<tr>
<td>certificate={value}</td>
<td><em>(For create request, optional when login_type=pkcert)</em> The passphrase X.509 certificate content.</td>
</tr>
<tr>
<td>require_cert={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
1) Create an MongoDB system record template and enter the login credentials you want to use for system created records.

2) Select the MongoDB system record template in the compliance option profile you want to use for discovery scans.

3) Launch your discovery scan. Your scan results will list the auto discovered instances.

4) List your MongoDB authentication records. For each system created record, you'll see the template associated with the record.

**Sample - Create MongoDB record - basic login**

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=create&title=API-mongodb-basic-login&username=joe_user&password=joe_password&ips=10.20.32.239&comment=mongo-basic-login&unix_conf_path=/etc/mongodb.conf&port=28020&ssl_verify=0&database_name=admin" "https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/" > file.xml
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-12T22:43:27Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125709</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

**Sample - Create MongoDB record, using SSL**

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d "action=create&title=API-mongodb-basic-login-with-ssl-verify1_hosts&use_username=mongo-admin&password=<PASSWORD>&ips=10.20.32.239&comments=mongo-basic-login-ssl_hosts&unix_conf_path=/opt/mongodb/" > file.xml
```

326
Scan Authentication
MongoDB Record

ify=1&hosts=abc123.s2012r2.lab.acme.com,abc123.s2008r2.lab.acme.com" 
"https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/" > file.xml

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM 
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-03-12T22:45:06Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125710</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

Sample - Create MongoDB record, using vault

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d 
"action=create&title=API-mongo-vault-CA_Access&ips=10.20.32.239&comme nts=mongo-CA-Access-vault_login&unix_conf_path=/opt/mongodb4.conf/&po rt=27010&login_type=vault&vault_type=CA Access 
Control&vault_id=166657&end_point_name=name&end_point_type=type&end_p oint_container=container&username=joe_user" 
"https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/" > file.xml

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM 
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-03-12T22:46:47Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125711</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
Sample - Create MongoDB Record for LDAP Authentication

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=create&title=Sample1&username=mlqa&password=<PASSWORD>&ips=10.20.32.107&comments=Creating through API v2.0&unix_conf_path=/etc/mongod3111.conf&port=28021&ssl_verify=0&database_name=admin&credential_type=external&cleartext=1" "https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb"
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-09-08T06:15:39Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>3052106</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Create MongoDB record - basic login and require_cert=1

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=create&title=mongo_auth_basic_cert&username=joe_user&password=<PASSWORD>&login_type=basic&ips=10.20.30.40&database_name=admin&port=27019&require_cert=1&unix_conf_path=/etc/mongod2.conf&ssl_verify=1&hosts=mlcent76mdb34.s2012r2.qualys.com' --header 'X-Requested-With: qweb' --header 'Authorization: Basic <token>' --header 'Content-Type: application/x-www-form-urlencoded' --data-urlencode 'certificate=-----BEGIN CERTIFICATE-----MIIErDCCApSgAwIBAgIBIDANBgkqhkiG9w0BAQUFADCBljEbMBkGA1UEAwwSU2Nhbm5lb29uZSBRQSBSb290IENBMRMwEQYDVQQIEwJVUzEwMBQGCCsGAQUFBzAALjEMMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEoAwIBAQSgAquS6T7dM7E/V91z7Hl5229sxhO8B7Xj9/0rtn6MTHvov518GnlG6DjixJ8Xh/5cR7JNkz4ruD6L7pI8x/Pa4W7500m7iF7z1rQ30S9W0jwxnB7XzqY4/h9js/Gv6Gc7mQ4sZz.o/4v4W6ZHr7q0RQ+kL2J1iA5D2G7aU1Wz7OZyW9xZlO9zWOOJ5UeU1eQom9s5qJmRj785q7ZrW6Ibq5f4jQyCKWj/9Lw33BO25hKula56pE3IY84H2y8DZoaoMx66JRoJF63sYp2EQRJ/Hupx8E5e4yGdnSwbF0dQad7uM9j6U3cQsG05uV3 Burgess5CJbKkeH73kT6wvYo1BZGyWt7yVR7ZN5laQ29sht23qzuy8Zv2d7N/0G/9eGjP9GnI6cBZ2JG0Rg7uWgCRey6kD1sGmU4sYXQ60m7pi5z3ZU9yOfB0zg8t+2YArP55fLk7G9kmDGb/9JGlniwG6Uq4d8gUgM0Gj2i/6p9vQ9Tz4+Wz6svv9fh2r+19J7O238lMv200R/Cq9X+PvT8nORyW2KQIc7i54h1/DVjGrn9K6ZUW49y8QzQxKv3reL4pvaXe34mLr+94p/9s+9Z7H0Lms/y41H1Imx24QYIg0BZPidaK9Tc+8nB6jM5rX6L7+A7d6H6I/M+b3cR7yu5y6VQ6ZfEGmNv5rE8h15tb74J5H9V/3a/tcvi8kYvVDolwX6Qgb7Hf9F//2020-09-08T06:15:39Z"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-09-08T06:15:39Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>3052106</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Sample - List MongoDB records

**API request:**

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d "action=list&details=All" "https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/" > file.xml
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_MONGODB_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/auth_mongodb_list_output.dtd">
<AUTH_MONGODB_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-09-12T22:42:45Z</DATETIME>
    <AUTH_MONGODB_LIST>
      <AUTH_MONGODB>
        <ID>125693</ID>
        <TITLE><![CDATA[API-mongo-basic-login]]></TITLE>
        <USERNAME><![CDATA[mongo-admin-name]]></USERNAME>
        <DATABASE><![CDATA[db-admin-name]]></DATABASE>
        <PORT>28020</PORT>
      </AUTH_MONGODB>
    </AUTH_MONGODB_LIST>
  </RESPONSE>
</AUTH_MONGODB_LIST_OUTPUT>
```
<UNIX_CONFIGURATION_FILE><![CDATA[/opt/mongodb/updated]]></UNIX_CONFIGURATION_FILE>
  <IP_SET>
    <IP>10.20.32.239</IP>
  </IP_SET>
<LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
<NETWORK_ID>0</NETWORK_ID>
<CREATED>
  <DATETIME>2017-09-12T20:22:09Z</DATETIME>
  ...

**DTDs for auth type “mongodb”**

*platform API server>/api/2.0/batch_return.dtd
*platform API server>/api/2.0/fo/auth/mongodb/auth_mongodb_list_output.dtd
MS Exchange Server

/api/2.0/fo/auth/ms_exchange/

[POST]

Create, update, list and delete MS Exchange Server authentication records. Compliance scans are supported (using PC).

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
</tbody>
</table>

Target Hosts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>(Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs to the IPs list for this record. Multiple IPs/ranges are comma separated.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>

Sample - Create MS Exchange Server record

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d
Scan Authentication
MS Exchange Server

"action=create&network_id=0&title=fordeltes&comments=editapicomment&ips=10.10.10.31"
"https://qualysapi.qualys.com/api/2.0/fo/auth/ms_exchange/

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-03-20T08:26:54Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>49029</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

Sample - List MS Exchange Server records
Use the new MS Exchange Server Authentication Record List API (/api/2.0/fo/auth/ms_exchange/?action=list) to list MS Exchange Server records.

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list&details=Basic"
"https://qualysapi.qualys.com/api/2.0/fo/auth/ms_exchange/

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_MS_EXCHANGE_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/ms_exchange/auth_ms_exchange_list_output.dtd">
<AUTH_MS_EXCHANGE_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2019-03-20T07:26:38Z</DATETIME>
    <AUTH_MS_EXCHANGE_LIST>
      <AUTH_MS_EXCHANGE>
        <ID>48050</ID>
        <TITLE><![CDATA[msexchange01]]></TITLE>
      </AUTH_MS_EXCHANGE>
    </AUTH_MS_EXCHANGE_LIST>
  </RESPONSE>
</AUTH_MS_EXCHANGE_LIST_OUTPUT>
<TITLE>
</TITLE>
<IP_SET>
  <IP>10.10.10.10</IP>
</IP_SET>
<NETWORK_ID>0</NETWORK_ID>
<CREATED>
  <DATETIME>2019-03-14T07:05:05Z</DATETIME>
  <BY>quays_sp1</BY>
</CREATED>
<LAST_MODIFIED>
  <DATETIME>2019-03-14T07:05:05Z</DATETIME>
</LAST_MODIFIED>
<COMMENTS>
  <![CDATA[msexchange]]>
</COMMENTS>
</AUTH_MS_EXCHANGE>
...

<AUTH_MS_EXCHANGE>
  <ID>49026</ID>
  <TITLE>
    <![CDATA[apicreate]]>
  </TITLE>
  <IP_SET>
    <IP>10.10.10.13</IP>
  </IP_SET>
  <NETWORK_ID>0</NETWORK_ID>
  <CREATED>
    <DATETIME>2019-03-19T11:46:23Z</DATETIME>
    <BY>quays_sp1</BY>
  </CREATED>
  <LAST_MODIFIED>
    <DATETIME>2019-03-19T11:56:57Z</DATETIME>
  </LAST_MODIFIED>
  <COMMENTS>
    <![CDATA[editapicomment]]>
  </COMMENTS>
</AUTH_MS_EXCHANGE>
</AUTH_MS_EXCHANGE_LIST>
</RESPONSE>
</AUTH_MS_EXCHANGE_LIST_OUTPUT>

DTDs for auth type "ms_exchange"

<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/ms_exchange/auth_ms_exchange_list_output.dtd
Sample: Update MS Exchange Server record

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=update&ids=49029&title=forupdate&comments=editwapicomment&ips=10.10.10.11" "https://qualysapi.qualys.com/api/2.0/fo/auth/ms_exchange/"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-03-20T08:29:48Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>49029</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample: Delete MS Exchange Server record (single)

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=delete&ids=49026" "https://qualysapi.qualys.com/api/2.0/fo/auth/ms_exchange/"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-03-20T07:56:00Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>49026</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Sample: Delete MS Exchange Server records (bulk)

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=delete&ids=49028,49029" "https://qualysapi.qualys.com/api/2.0/fo/auth/ms_exchange/

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-03-20T08:31:35Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID_RANGE>49028-49029</ID_RANGE>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
MS SQL Record
/api/2.0/fo/auth/ms_sql/

[POST]

Create, update, list and delete MS SQL Server authentication records. Compliance scans are supported (using PC).

Requirement - You must configure authentication credentials on target hosts.

Download Qualys User Guide - MS SQL Server 2000 Authentication (.pdf)

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) User defined comments. Maximum 1999 characters.</td>
</tr>
</tbody>
</table>

Login credentials

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The user account to be used for authentication. May include 1-128 characters.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required to create record, optional to update record) The password corresponding to the user account defined in the record for authentication. May include 1-128 characters.</td>
</tr>
<tr>
<td>db_local={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
### Parameter | Description
---|---
`windows_domain={value}`  | (Required when `db_local`=0, otherwise invalid)
The domain name where the login credentials are stored when the login credentials are for a Microsoft Windows operating system account that is associated with a MS SQL Server database account. The domain name may include 1-256 characters (ascii).
For an update request when the credentials for the record are for a Microsoft Windows account (`db_local`=0) and you want to change the record to use credentials for a MS SQL Server account (`db_local`=1) note the following. You must set `windows_domain=''` (the empty string) to clear the current parameter setting.

`auth_os_type={unix|windows}`  | (Optional when `db_local`=1) Specify "unix" when the OS type is Unix and "windows" when the OS type is Windows.

`mssql_unix_insta_path={value}`  | (Optional when `auth_os_type`=unix) Specify the path to the MS SQL Server instance directory on Unix hosts. Sample value: `/var/opt/mssql`

`mssql_unix_conf_path={value}`  | (Optional when `auth_os_type`=unix) Specify the path to the MS SQL Server configuration file on Unix hosts. Sample value: `/var/opt/mssql/mssql.conf`

`instance={value}`  | (Optional to create or update record for Windows, Required to create record for Unix and Optional to update record for Unix) The name of the database instance to be scanned. This is the instance name assigned to the TCP/IP port. Important: This is not the host name that is assigned to the MS SQL Server instance name (see "MS SQL Server Instance Name" in the Qualys online help for information). The instance name may include a maximum of 128 characters (ascii).
If the `instance` parameter is not specified for Windows, the instance name is set to "MSSQLSERVER".
These parameters are mutually exclusive: `instance` and `auto_discover_instances`=1.

`auto_discover_instances={0|1}`  | (Optional when `auth_os_type`=windows) Set `auto_discover_instances`=1 and we’ll find all MS SQL Server instance names on each Windows host. Note that Windows authentication is required in order for us to auto discover instance names. Set up Windows authentication records for the hosts running MS SQL Servers.
These parameters are mutually exclusive: `instance` and `auto_discover_instances`=1.
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>database={value}</td>
<td>(Optional to create or update record) The database name of the database to be scanned. The database name may contain a maximum of 128 characters. For a create request, if the database name is unspecified, the database name is set to “master”.</td>
</tr>
<tr>
<td>auto_discover_databases={0</td>
<td>1}</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Required to create record, optional to update record) The port number assigned to the database instance to be scanned. To create a record you must specify one of these parameters: port or auto_discover_ports=1. These parameters are mutually exclusive.</td>
</tr>
<tr>
<td>auto_discover_ports={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

#### Target Hosts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>You may enter a combination of IPs and IP ranges to identify compliance hosts. Multiple entries are comma separated. (Optional to update record) Overwrites (replaces) the IP list for the authentication record. The IPs you specify are added and any existing IPs are removed. For create request, it is required to specify either this parameter or member_domain parameter. For update request, this parameter and the add_ips or remove_ips or member_domain parameter cannot be specified in the same request.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) You may enter a combination of IPs and IP ranges to identify compliance hosts. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter is used to update an existing IP list in an existing authentication record. Specifies one or more IP addresses to add to the IP list for the authentication record.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips or member_domain parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional for update request only) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips or member_domain parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional and only valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
<tr>
<td>member_domain={value}</td>
<td>(Optional and only valid for Windows) Defines the domain of the MS SQL server for the authentication record.</td>
</tr>
<tr>
<td></td>
<td>For create request, it is required to specify either this parameter or ips or add_ips parameter.</td>
</tr>
<tr>
<td></td>
<td>For update request, this parameter and the ips or add_ips or remove_ips parameter cannot be specified in the same request.</td>
</tr>
</tbody>
</table>

#### Protocols (Windows only)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>kerberos={0</td>
<td>1}</td>
</tr>
<tr>
<td>ntlmv2={0</td>
<td>1}</td>
</tr>
<tr>
<td>ntlmv1={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
Sample Create MS SQL Record for Unix

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=create&title=MSSQL_UNIX&username=root&password=<PASSWORD>&db_local=1&ips=10.10.10.10&auto_discover_ports=1&auto_discover_databases=1&auth_os_type=unix&instance=mssql&msql_unix_conf_path=/var/opt/mssql/mssql.conf&msql_unix_insta_path=/var/opt/mssql" "https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/
```

**Response:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-05-17T08:26:31Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>103473</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - List record for Windows using member domain

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=list&echo_request=1&ids=13907" "https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_MS_SQL_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/auth_ms_sql_list_output.dtd">
<AUTH_MS_SQL_LIST_OUTPUT>
  <REQUEST>
    <DATETIME>2017-09-20T05:34:37Z</DATETIME>
    <USER_LOGIN>user_john</USER_LOGIN>
    <RESOURCE>
      https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/
    </RESOURCE>
    <PARAM_LIST>
```

341
<PARAM>
  <KEY>action</KEY>
  <VALUE>list</VALUE>
</PARAM>

<PARAM>
  <KEY>echo_request</KEY>
  <VALUE>1</VALUE>
</PARAM>

<PARAM>
  <KEY>ids</KEY>
  <VALUE>13907</VALUE>
</PARAM>

</PARAM_LIST>

</REQUEST>

<RESPONSE>
  <DATETIME>2017-09-20T05:34:37Z</DATETIME>

  <AUTH_MS_SQL_LIST>
    <AUTH_MS_SQL>
      <ID>13907</ID>
      <TITLE><![CDATA[mssqlvt4]]></TITLE>
      <USERNAME><![CDATA[administrator]]></USERNAME>
      <NTLM_V2>1</NTLM_V2>
      <KERBEROS>1</KERBEROS>
      <INSTANCE><![CDATA[MSSQLSERVER]]></INSTANCE>
      <DATABASE><![CDATA[master]]></DATABASE>
      <PORT>8012</PORT>
      <DB_LOCAL>1</DB_LOCAL>

      <MEMBER_DOMAIN><![CDATA[sitedomain.com]]></MEMBER_DOMAIN>
      <NETWORK_ID>0</NETWORK_ID>

      <CREATED>
        <DATETIME>2017-09-20T05:26:31Z</DATETIME>
        <BY>user_john</BY>
      </CREATED>

      <LAST_MODIFIED>
        <DATETIME>2017-09-20T05:26:31Z</DATETIME>
      </LAST_MODIFIED>

      <COMMENTS><![CDATA[authcreated]]></COMMENTS>
    </AUTH_MS_SQL>
  </AUTH_MS_SQL_LIST>
</RESPONSE>

</AUTH_MS_SQL_LIST_OUTPUT>
Sample - Create record for Windows using member domain

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=create&title=mssqlvt4&username=administrator&password=abc123&db_local=1&port=8012&member_domain=sitedomain.com&echo_request=1&comments=authcreated&instance=MSSQLSERVER&database=master" "https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/"

XML output:

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
<REQUEST>
<DATETIME>2018-03-20T05:26:31Z</DATETIME>
<USER_LOGIN>user_john</USER_LOGIN>
<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/</RESOURCE>
<PARAM_LIST>
<PARAM>
<Key>action</Key>
<Value>create</Value>
</PARAM>
<PARAM>
<Key>title</Key>
<Value>mssqlvt4</Value>
</PARAM>
<PARAM>
<Key>username</Key>
<Value>administrator</Value>
</PARAM>
<PARAM>
<Key>password</Key>
<Value>abc123</Value>
</PARAM>
<PARAM>
<Key>db_local</Key>
<Value>1</Value>
</PARAM>
<PARAM>
<Key>port</Key>
<Value>8012</Value>
</PARAM>
</PARAM_LIST>
</REQUEST>
</BATCH_RETURN>
Scan Authentication
MS SQL Record

<KEY>member_domain</KEY>
<VALUE>sitedomain.com</VALUE>
</PARAM>
<PARAM>
  <KEY>echo_request</KEY>
  <VALUE>1</VALUE>
</PARAM>
<PARAM>
  <KEY>comments</KEY>
  <VALUE>authcreated</VALUE>
</PARAM>
<PARAM>
  <KEY>instance</KEY>
  <VALUE>MSSQLSERVER</VALUE>
</PARAM>
<PARAM>
  <KEY>database</KEY>
  <VALUE>master</VALUE>
</PARAM>
</PARAM_LIST>
</REQUEST>
</RESPONSE>

<DATETIME>2018-03-20T05:26:31Z</DATETIME>
<BATCH_LIST>
<BATCH>
<TEXT>Successfully Created</TEXT>
<ID_SET>
  <ID>13907</ID>
</ID_SET>
</BATCH>
</BATCH_LIST>
</RESPONSE>
</BATCH_RETURN>

**Sample - Update record for Windows using member domain**

**API request:**
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=update&echo_request=1&ids=13907&member_domain=webdomain.com"
"https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/
```

**XML output:**
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM

344```
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <REQUEST>
    <DATETIME>2018-03-20T05:37:13Z</DATETIME>
    <USER_LOGIN>user_john</USER_LOGIN>
    <RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/</RESOURCE>
    <PARAM_LIST>
      <PARAM>
        <KEY>action</KEY>
        <VALUE>update</VALUE>
      </PARAM>
      <PARAM>
        <KEY>echo_request</KEY>
        <VALUE>1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>ids</KEY>
        <VALUE>13907</VALUE>
      </PARAM>
      <PARAM>
        <KEY>member_domain</KEY>
        <VALUE>webdomain.com</VALUE>
      </PARAM>
    </PARAM_LIST>
  </REQUEST>
  <RESPONSE>
    <DATETIME>2018-03-20T05:37:13Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET><ID>13907</ID>
          </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

DTDs for auth type "ms_sql"
<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/ms_sql/auth_ms_sql_list_output.dtd
MySQL Record

/api/2.0/fo/auth/mysql/

[POST]

Create, update, list and delete MySQL records for authenticated scans of MySQL Server instances. Vulnerability and compliance scans are supported (using VM, PC).

Requirement - You must configure authentication credentials on target hosts.

Download Qualys User Guide - MySQL Authentication (.zip)

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
<tr>
<td></td>
<td>- If unspecified (or ssl_verify=0), Qualys scanners authenticate with MySQL Servers that don't use SSL or MySQL servers that use SSL. However, in the SSL case, the server SSL certificate verification will be skipped.</td>
</tr>
<tr>
<td></td>
<td>- If ssl_verify=1, the Qualys scanners will only send a login request after verifying that a connection the MySQL server uses SSL, the server SSL certificate is valid and matches the scanned host.</td>
</tr>
<tr>
<td>hosts={value}</td>
<td>(Optional to create or update record) A list of FQDNs for the hosts that correspond to all host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed. Multiple hosts are comma separated.</td>
</tr>
</tbody>
</table>
**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>database={value}</td>
<td>(Required to create, optional to update record) The database name to authenticate to. Specify a valid MySQL database name.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Required to create, optional to update record) The port the database name is running on.</td>
</tr>
<tr>
<td>windows_config_file={value}</td>
<td>(Optional to create or update record) The path to the Windows MySQL config file. Access to this config file is required to run certain checks on Windows hosts. Note: You must specify either windows_config_file or unix_config_file depending on the host OS.</td>
</tr>
<tr>
<td>unix_config_file={value}</td>
<td>(Optional) Name of the client (Consultant type subscriptions). Note: You must specify either windows_config_file or unix_config_file depending on the host OS.</td>
</tr>
<tr>
<td>client_cert={value}</td>
<td>(Optional to create or update record) PEM-encoded X.509 certificate. Specify if certificate authentication is required by your server to establish an SSL connection.</td>
</tr>
<tr>
<td>client_key={value}</td>
<td>(Optional to create or update record) PEM-encoded RSA private key. Specify if certificate authentication is required by your server to establish an SSL connection.</td>
</tr>
</tbody>
</table>

**Login credentials**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>login_type={basic</td>
<td>vault}</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The username of the account to be used for authentication. If password is specified this is the username of a MySQL account. If login_type=vault is specified, this is the username of a vault account.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required to create record, optional to update record) The password to be used for authentication to MySQL server. Maximum 100 characters (ascii).</td>
</tr>
</tbody>
</table>

**Vault**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vault_type={value}</td>
<td>(Required only when action=create and login_type=vault) The vault to be used for authentication. See Vault Support matrix.</td>
</tr>
<tr>
<td>vault_id={value}</td>
<td>(Required only when action=create and login_type=vault) The ID of the vault you want to use.</td>
</tr>
<tr>
<td>{vault parameters}</td>
<td>(Required only when action=create and login_type=vault) Vault specific parameters required depend on the vault type you've selected. See Vault Definition.</td>
</tr>
</tbody>
</table>

**Target Hosts**
Scan Authentication
MySQL Record

**Sample - List MySQL record**

You'll see vault information in the XML output when you list MySQL authentication records with vaults.

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=list&ids=284212" "https://qualysapi.qualys.com/api/2.0/fo/auth/mysql/
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_MYSQL_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/mysql/auth_mysql_list_output.dtd">
<AUTH_MYSQL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-07-17T17:09:18Z</DATETIME>
    <AUTH_MYSQL_LIST>
      <AUTH_MYSQL>
        <ID>284212</ID>
        <TITLE><![CDATA[api-Thycotic Secret Server_tss]]></TITLE>
        <USERNAME><![CDATA[test_tss]]></USERNAME>
        <DATABASE><![CDATA[mysql]]></DATABASE>
        <PORT>22</PORT>
        <HOSTS>
          <HOST><![CDATA[www.test.com]]></HOST>
        </HOSTS>
      </AUTH_MYSQL>
    </AUTH_MYSQL_LIST>
  </RESPONSE>
</AUTH_MYSQL_LIST_OUTPUT>
```

**Parameter** | **Description**
---|---
ips={value} | (Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.
| (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.
add_ips={value} | (Optional to update record) Add IPs to the IPs list for this record. Multiple IPs/ranges are comma separated.
remove_ips={value} | (Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.
| This parameter and the ips or member_domain parameter cannot be specified in the same request.
network_id={value} | (Optional and valid when the networks feature is enabled) The network ID for the record.

---

ips={value} (Required to create record) The IP address(es) the server will log into using the record's credentials. Multiple entries are comma separated.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs to the IPs list for this record. Multiple IPs/ranges are comma separated.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>
Sample - Create new MySQL record

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=create&title=NewMySQLRecord&username=USERNAME&password=<PASSWORD>&ips=10.10.31.84&echo_request=1&windows_config_file=c:\mysql\my.ini&port=22&database=mysql" "https://qualysapi.qualys.com/api/2.0/fo/auth/mysql/"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-07-27T17:02:23Z</DATETIME>
  </RESPONSE>
</AUTH_MYSQL_OUTPUT>
</AUTH_MYSQL_LIST_OUTPUT>

Scan Authentication
MySQL Record
Sample - Create MySQL record, using vault

API request:


XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-07-27T17:14:57Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>291734</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
Sample - Update MySQL record

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=update&ids=137296922&password=<NEWPASSWORD>"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mysql/"
```

XML output:
```xml
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-01-23T17:14:28Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>137296922</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Update vault details in MySQL record

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=update&ids=272380&ips=10.10.10.19&username=USERNAME&title=
NewMySQLRecord&ssl_verify=0&login_type=vault&vault_type=CyberArk
PIM
&vault_id=248308&folder=folder&file=file&hosts=www.qualys.com
&comments=test comments updated"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mysql/"
```

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>284212</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
DTDs for auth type "mysql"
<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/mysql/auth_mysql_list_output.dtd
Scan Authentication

Neo4j Record

/api/2.0/fo/auth/neo4j/

[POST]

Create, update, list and delete Neo4j authentication records. Compliance scans are supported (using PC and SCA). User permissions for this API are the same as other authentication record APIs.

Requirement - You must configure authentication credentials on target hosts.

Download Qualys User Guide - Neo4j Authentication

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required to create record) Enter a combination of IPs and IP ranges to identify compliance hosts. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional and valid only to update record) Add IPs to the IP list for an existing record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional and valid only to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>database={value}</td>
<td>(Optional to create or update record) The database name of the Neo4j database to be scanned. The database name may contain a maximum of 255 multi-byte characters.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Required to create record, optional to update record) The port number assigned to the database instance to be scanned.</td>
</tr>
<tr>
<td>login_type={basic</td>
<td>vault}</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The username to be used for authentication to Neo4j.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required to create record) When login_type=basic, specify the password to be used for authentication to Neo4j. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>vault_id={value}</td>
<td>Required if login_type=vault. The ID of the vault to be used to retrieve the password for login.</td>
</tr>
<tr>
<td>vault_type={value}</td>
<td>Required if login_type=vault. The third party vault to be used to retrieve the password for login. Certain vaults support this capability.</td>
</tr>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
<tr>
<td></td>
<td>- If ssl_verify=0, the Qualys scanners authenticate with In Servers that don’t use SSL or Neo4j servers that use SSL. However, in the SSL case, the server SSL certificate verification will be skipped.</td>
</tr>
<tr>
<td></td>
<td>- If unspecified (or ssl_verify=1), the Qualys scanners will only send a login request after verifying that a connection to the Neo4j server uses SSL, the server SSL certificate is valid and matches the scanned host.</td>
</tr>
<tr>
<td>hosts={value}</td>
<td>(Required only when ssl_verify is enabled) A list of FQDNs for the hosts that correspond to all host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed. Multiple hosts are comma separated.</td>
</tr>
<tr>
<td>neo4j_version={value}</td>
<td>(Optional) Specifies the Neo4j version. Only Neo4j 3.x version is supported at this time. Valid value is “neo4j 3.x” (case insensitive). When unspecified, Neo4j 3.x is used.</td>
</tr>
<tr>
<td>unix_base_path={value}</td>
<td>(Optional) The base path for Neo4j on your Unix hosts. Sample value: /opt/neo4j-enterprise-3.5.16/</td>
</tr>
<tr>
<td></td>
<td>Instead of specifying the path information, you can choose to auto discover the base and configuration paths by specifying neo4j_auto_path=1.</td>
</tr>
<tr>
<td>unix_conf_path={value}</td>
<td>(Optional) The path to the Neo4j configuration file on your Unix hosts. Sample value: /opt/neo4j-enterprise-3.5.16/conf/neo4j.conf</td>
</tr>
<tr>
<td></td>
<td>Note that the configuration file must be in the same location for all hosts (IPs) included in the record. Instead of specifying path information, you can choose to auto discover the base and configuration paths by specifying neo4j_auto_path=1.</td>
</tr>
<tr>
<td>neo4j_auto_path={0</td>
<td>1}</td>
</tr>
<tr>
<td></td>
<td>When neo4j_auto_path=1 (true) we will auto discover the base and configuration paths for Neo4j on your Unix hosts.</td>
</tr>
</tbody>
</table>
Sample: Create Neo4j Record

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=create&title=neo4j-recordAuth
Record&username=root&password=<PASSWORD>&database=graph.db&port=7687&ips=1.1.14&unix_conf_path=/opt/neo4j-enterprise-
3.5.16/conf/neo4j.conf&unix_base_path=/opt/neo4j-enterprise-
3.5.16/&neo4j_version=neo4j 3.x&neo4j_auto_path=0"
"https://qualysapi.qualys.com/api/2.0/fo/auth/neo4j/"
```

Response:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-03-15T11:56:08Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>101430</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample: Update Neo4j Record

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=update&title=Neo4j Auth Record
&username=root&password=<PASSWORD>&database=graph.db&port=7689&ips=1.1.1
.1&ids=101430&unix_conf_path=/opt/neo4j-enterprise-
3.5.16/conf/neo4j.conf&unix_base_path=/opt/neo4j-enterprise-
3.5.16/&neo4j_version=neo4j 3.x&neo4j_auto_path=0"
"https://qualysapi.qualys.com/api/2.0/fo/auth/neo4j/"
```

Response:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-03-15T11:56:08Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>101430</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Sample - List Neo4j Record

You’ll see Neo4j record IDs in the output when you have Neo4j records in your account.

API request:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list" "https://qualysapi.qualys.com/api/2.0/fo/auth/neo4j/
```

Response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_NEO4J_LIST_OUTPUT SYSTEM
  "https://qualysapi.qualys.com/api/2.0/fo/auth/neo4j/auth_neo4j_list_output.dtd">
<AUTH_NEO4J_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-05-24T10:23:14Z</DATETIME>
    <AUTH_NEO4J_LIST>
      <AUTH_NEO4J>
        <ID>4815851</ID>
        <TITLE><![CDATA[Neo4j Sample]]></TITLE>
        <USERNAME><![CDATA[root]]></USERNAME>
        <DATABASE><![CDATA[alpha]]></DATABASE>
        <PORT>123</PORT>
        <SSL_VERIFY><![CDATA[0]]></SSL_VERIFY>
        <IP_SET>
          <IP>10.10.10.10</IP>
          <IP>10.10.10.20</IP>
        </IP_SET>
        <UNIX_CONF_PATH><![CDATA[/opt/neo4j-enterprise-3.5.16/conf/neo4j.conf]]></UNIX_CONF_PATH>
        <UNIX_BASE_PATH><![CDATA[/opt/neo4j-enterprise-3.5.16/]]></UNIX_BASE_PATH>
        <VERSION><![CDATA[Neo4j 3.x]]></VERSION>
        <AUTO_PATH><![CDATA[0]]></AUTO_PATH>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2021-05-24T10:46:38Z</DATETIME>
          <BY>joe_user</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2021-05-24T10:48:36Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_NEO4J>
    </AUTH_NEO4J_LIST>
  </RESPONSE>
</AUTH_NEO4J_LIST_OUTPUT>
```
Sample: Delete Neo4j Records

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=delete&ids=4620768"
"https://qualysapi.qualys.com/api/2.0/fo/auth/neo4j/"

Response:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-04-01T13:12:51Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>4620768</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

DTDs for auth type “neo4j”

<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/neo4j/auth_neo4j_list_output.dtd
**Nginx Record**

(api/2.0/fo/auth/nginx/)

[POST]

Create, update, list and delete Nginx authentication records. Compliance scans are supported (using PC and SCA). User permissions for this API are the same as other authentication record APIs.

Requirement - You must configure authentication credentials on target hosts.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required to create record) Enter a combination of IPs and IP ranges to identify compliance hosts. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional and valid only to update record) Add IPs to the IP list for an existing record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional and valid only to update record) IPs to be removed from your record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>unix_bin_path={value}</td>
<td>(Optional) Absolute path of the Nginx binary file location.</td>
</tr>
<tr>
<td>unix_conf_path={value}</td>
<td>(Optional) The path to the Nginx configuration file on your Unix hosts.</td>
</tr>
<tr>
<td>unix_prefix_path</td>
<td>(Optional) The path to the Nginx configuration file on your Unix hosts.</td>
</tr>
</tbody>
</table>

**Sample: Create Nginx Record**

**API request:**

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=create&ips=1.2.3.4&title=API_Nginx&unix_bin_path=/usr/local/nginx/sbin/nginx&unix_conf_path=/usr/local/nginx/conf/nginx.conf &unix_prefix_path=/usr/local/nginx" "https://qualysapi.qualys.com/api/2.0/fo/auth/nginx/"
```
Response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-08-13T11:36:30Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>1157719</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample: Update Nginx Record

API request:
```
```

Response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-08-03T03:15:35Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>229028</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample: List the Nginx Records

You’ll see Nginx record IDs in the output when you have Nginx records in your account.

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list" "https://qualysapi.qualys.com/api/2.0/fo/auth/nginx"
```
Response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_RECORDS_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/nginx/auth_nginx_list_output.dtd">
<AUTH_NGINX_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-08-02T13:57:09Z</DATETIME>
    <AUTH_NGINX_LIST>
      <AUTH_NGINX>
        <ID>228028</ID>
        <TITLE><![CDATA[Nginx second]]></TITLE>
        <IP_SET>
          <IP>10.11.12.13</IP>
        </IP_SET>
        <UNIX_BIN_PATH><![CDATA[/usr/local/nginx/sbin/nginx]]></UNIX_BIN_PATH>
        <UNIX_CONF_PATH><![CDATA[/usr/local/nginx/conf/nginx.conf]]></UNIX_CONF_PATH>
        <UNIX_PREFIX_PATH><![CDATA[/usr/local/nginx]]></UNIX_PREFIX_PATH>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2021-07-29T06:15:12Z</DATETIME>
          <BY>joe_user</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2021-07-29T07:20:17Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_NGINX>
    </AUTH_NGINX_LIST>
    <GLOSSARY>
      <USER_LIST>
        <USER>
          <USER_LOGIN>joe_user</USER_LOGIN>
          <FIRST_NAME>Joe</FIRST_NAME>
          <LAST_NAME>User</LAST_NAME>
        </USER>
      </USER_LIST>
    </GLOSSARY>
  </RESPONSE>
</AUTH_NGINX_LIST_OUTPUT>
```

Sample: Delete Nginx Records

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=delete&ids=5146728,5146726"
Response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-08-27T11:38:07Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>5146726</ID>
          <ID>5146728</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

**DTDs for auth type “nginx”**

- `<platform API server>/api/2.0/batch_return.dtd`
- `<platform API server>/api/2.0/fo/auth/neo4j/auth_nginx_list_output.dtd`
Oracle Record

/api/2.0/fo/auth/oracle/

[POST]

Create, update, list and delete Oracle records and Oracle system record templates for authenticated scans of Oracle instances. Vulnerability and compliance scans are supported (using VM, PC).

How it works - During scanning we’ll authenticate to one or more instances on a single host using all Oracle records in your account. For compliance scans, you can scan multiple Oracle instances on a single host and port combination. Looking for more help? Search for “Oracle Use Cases” in Qualys online help.

System created authentication records supported - You can allow the system to create Oracle authentication records for auto discovered instances and scan them. This is supported for Unix installations only. To enable this feature, you must first create Oracle System Record Templates using the is_template input parameter and specifying login credentials. See System created Oracle records.

Requirement - You must configure login credentials on target hosts before scanning.

Download Qualys User Guide - Oracle Authentication for VM Scans (.zip)
Download Qualys User Guide - Oracle Authentication for Compliance Scans (.zip)

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record, optional to update record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>is_template={0</td>
<td>1}</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>status=[0</td>
<td>1]</td>
</tr>
<tr>
<td>save_as_user_auth=[0</td>
<td>1]</td>
</tr>
</tbody>
</table>

**Login credentials**

| login_type=[basic|vault]            | (Optional) The login type is basic by default. You can choose vault (for vault based authentication).                                                                                                      |
| username={value}                  | (Required to create record, optional to update record) The user account to be used for authentication to the Oracle database. The username may include 1-31 characters (ascii).                                             |
| password={value}                  | (Required to create record, optional to update record) The password corresponding to the user account defined in the record for authentication. Maximum 100 characters (ascii).                                             |
| vault_type={value}                | (Required if login_type=vault) The third party vault to be used to retrieve the password for login. Certain vaults support this capability. See Vault Support matrix                                                                 |
| vault_id={value}                  | (Required to create record, optional to update record) The vault ID from where you want to retrieve the password. Certain vaults support this capability.                                                                 |
| {vault parameters}               | (Required to create record when login_type=vault) Vault specific parameters required depend on the vault type you’ve selected. See Vault Definition                                                                 |
| sid={value}                      | (Optional to create or update record) The Oracle System ID (SID) that identifies the database instance to be authenticated to. To create a record sid or servicename is required. The parameters sid and servicename cannot be specified in the same request. |
| servicename={value}              | (Optional to create or update record) The Oracle service name that identifies the database instance to be authenticated to. A maximum of 30 characters may be specified. The parameters sid and servicename cannot be specified in the same request. |
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>port={value}</td>
<td>(Optional to create record) The port number that the Oracle database instance is running on. When not specified, the &quot;All Ports&quot; option is used and the scanning engine will authenticate to the database instance on each port that the Oracle service is detected on. <strong>Forts used for Oracle authentication</strong>&lt;br&gt;These parameters are mutually exclusive: instance and auto_discover_instances=1.</td>
</tr>
<tr>
<td>is_cdb={0</td>
<td>1}</td>
</tr>
<tr>
<td>pc_only={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

### Target Hosts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>remove_ips=(value)</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id=(value)</td>
<td>(Optional and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>

#### OS Parameters Windows

OS Parameters are used for compliance scans only.

- **perform_windows_os_checks** = [0][1] *(Optional) Specify 1 to perform OS-dependent compliance checks for the Oracle technology during Windows authenticated compliance scans. These checks are assigned to the control category “Database Settings” in the sub-category “DB OS-dependent Controls”. |

- **win_ora_home_name** = (value) *(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The Windows Oracle Home name. Example: OraHome1 |

- **win_ora_home_path** = (value) *(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The Windows Oracle Home path. Example: c:\Program Files\Oracle\10 |

- **win_init_ora_path** = (value) *(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The pathname to the Windows init(SID).ora file. Example: c:\Program Files\oracle\dbs\initORA10.ora |

- **win_spfile_ora_path** = (value) *(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The pathname to the Windows spfile(SID).ora file. Example: c:\Program Files\oracle\network\admin\spfileORA10.ora |

- **win_listener_ora_path** = (value) *(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The pathname to the Window listener.ora file. Example: c:\Program Files\oracle\network\admin\listener.ora |

- **win_sqlnet_ora_path** = (value) *(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The pathname to the Windows sqlnet.ora file. Example: c:\Program Files\oracle\network\admin\sqlnet.ora |

- **win_tnsnames_ora_path** = (value) *(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The pathname to the Windows tnsnames.ora file. Example: c:\ProgramFiles\oracle\network\admin\tnsnames.ora |

#### OS Parameters Unix

OS Parameters are used for compliance scans only.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`perform_unix_os_checks={0</td>
<td>1}`</td>
</tr>
<tr>
<td>`perform_unix_opatch_checks={0</td>
<td>1}`</td>
</tr>
<tr>
<td><code>unix_ora_home_path={value}</code></td>
<td>(Required if <code>perform_unix_os_checks=1</code> and/or <code>perform_unix_opatch_checks=1</code> is specified; otherwise invalid) The Unix Oracle Home path. Example: <code>/usr/opt/oracle/10</code></td>
</tr>
<tr>
<td><code>unix_init_ora_path={value}</code></td>
<td>(Required if <code>perform_unix_os_checks=1</code> and/or <code>perform_unix_opatch_checks=1</code> is specified; otherwise invalid) The pathname to the Unix init(SID).ora file. Example: <code>/usr/opt/oracle/dbs/initORA10.ora</code></td>
</tr>
<tr>
<td><code>unix_spfile_ora_path={value}</code></td>
<td>(Required if <code>perform_unix_os_checks=1</code> and/or <code>perform_unix_opatch_checks=1</code> is specified; otherwise invalid) The pathname to the Unix spfile(SID).ora file. Example: <code>/usr/opt/oracle/network/admin/spfileORA10.ora</code></td>
</tr>
<tr>
<td><code>unix_listener_ora_path={value}</code></td>
<td>(Required if <code>perform_unix_os_checks=1</code> and/or <code>perform_unix_opatch_checks=1</code> is specified; otherwise invalid) The pathname to the Unix listener.ora file. Example: <code>/usr/opt/oracle/network/admin/listener.ora</code></td>
</tr>
<tr>
<td><code>unix_sqlnet_ora_path={value}</code></td>
<td>(Required if <code>perform_unix_os_checks=1</code> and/or <code>perform_unix_opatch_checks=1</code> is specified; otherwise invalid) The pathname to the Unix sqlnet.ora file. Example: <code>/usr/opt/oracle/network/admin/sqlnet.ora</code></td>
</tr>
</tbody>
</table>
Ports used for Oracle authentication

The “All Ports” option is used when the **port** parameter is not specified (the default). You may only create one Oracle record with this setting for each host. When All Ports is defined the scanning engine uses the credentials in the record to attempt authentication to the database instance (SID or service name) when a port-specific record does not exist. The scanning engine will authenticate to the database instance on each port the Oracle service is detected on.

A single port is used when the **port** parameter is specified (e.g. `port=1521`). The same port number cannot be entered in multiple Oracle records for the same host, unless the compliance module is enabled and `pc_only=1` is specified.

How it works - When the scanning engine detects an Oracle instance on a host, it first checks to see if you have an authentication record with the database instance and port specified. If you have a port-specific record, then it uses the credentials in that record to attempt authentication to the database instance. If a port-specific record does not exist (or if authentication fails), then the scanning engine checks to see if you have an authentication record set to “All Ports” for the host and uses the credentials in that record to attempt authentication to the database instance.

System created Oracle records

When we auto discover Oracle instances, we’ll discover the target configuration for each instance but not the login credentials. We’ve introduced a new configuration called “Oracle System Record Template” that you’ll use to provide Oracle login credentials for system created records. You’ll create the system record template and then select it in the option profile used for discovery scans. The template is linked automatically to the system created records created as a result of the scan.

Benefits

- We’ll auto discover Oracle instances on each scanned host and create authentication records for those instances. We support auto discovery and system record creation for Oracle instances running on Unix platforms. Make sure you have Unix authentication records in your account for hosts running Oracle.

- When we create Oracle authentication records for discovered instances, we’ll insert the credentials from the Oracle system record template you selected in the option profile.
- You can easily rotate Oracle passwords. Simply edit the credentials in the Oracle system record template and all Oracle records linked to the template will be updated to use the new credentials with no additional scan or action by you.

- You can edit individual Oracle system created records and save them as user created. This allows you to change the credentials for individual records without changing the credentials for all records associated with a template.

**How it works**

Here’s the basic flow for Oracle instance discovery and auto record creation. Note - We support auto discovery and system record creation for Oracle instances running on Unix platforms. Make sure you have Unix authentication records in your account for hosts running Oracle.

1) Create an Oracle system record template and enter the login credentials you want to use for system created records.

2) Select the Oracle system record template in the compliance option profile you want to use for discovery scans.

Note: You can configure Oracle system record template for Multitenant Container Database. Select the “Is CDB” option to enable your Oracle database system created authentication records in Multitenant Container Database.

3) Launch your discovery scan. Your scan results will list the auto discovered instances.

4) List your Oracle authentication records. For each system created record, you’ll see the template associated with the record.

**Sample create Oracle system record template**

This sample creates an Oracle system record template by using is_template=1.

**API request:**

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=create&is_template=1&title=OracleRecordTemplate&username=OracleUser&password=PASSWORD" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-04-23T18:43:59Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>2237956</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
DTDs for auth type “oracle”
<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/oracle/auth_oracle_list_output.dtd
Oracle Listener Record

/api/2.0/fo/auth/oracle_listener/

[POST]

Create, update, list and delete Oracle Listener records for authenticated scans of Oracle Listener databases. Vulnerability scans are supported (using VM).

Oracle Listener records are used to connect to Oracle TNS Listeners in order to enumerate information about databases behind the Oracle Listeners. When authentication is successful and databases behind the Listener are discovered, the QID 19225 “Retrieved Oracle Database Name” is returned in the scan results. This is an information gathered check that lists the names of the databases discovered behind the Listener. This information is useful if you want to create Oracle authentication records on those databases and need the Oracle System IDs (SIDs).

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=[action]</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request=[0</td>
<td>1]</td>
</tr>
<tr>
<td>ids=[value]</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title=[value]</td>
<td>(Required to create record, optional to update record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments=[value]</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>password=[value]</td>
<td>(Required to create record, optional to update record) Specifies a password for authentication to target hosts. If more than one Listener is detected on the same host, then the same password is attempted on each Listener. Maximum 100 characters (ascii).</td>
</tr>
</tbody>
</table>
### Target Hosts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>(Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>

### DTDs for auth type "oracle_listener"

```xml
<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/oracle_listener/auth_oracle_listener_list_output.dtd
```
Oracle WebLogic Server Record

/api/2.0/fo/auth/oracle_weblogic/

[POST]

Create, update, list and delete Oracle WebLogic records for authenticated scans of Oracle WebLogic Server instances. Vulnerability and compliance scans are supported (using VM, PC).

What you’ll need:
- We support these technologies: Oracle WebLogic Server 11g and Oracle WebLogic Server 12c
- Unix authentication is required so you’ll need a Unix record for each host running an Oracle WebLogic Server

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required for update request; invalid for create request) The IDs of the Oracle WebLogic Server authentication records that you want to update. Multiple IDs are comma separated</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>installation_path={value}</td>
<td>(Required to create record, optional to update record) The directory where the Oracle WebLogic Server is installed (i.e. Home directory). Example: /u01/app/oracle/middleware</td>
</tr>
<tr>
<td>auto_discover={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
## Sample - Create WebLogic record, no auto discover

### API request:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=create&installation_path=/u01/app/oracle&auto_discover=0&domain=www.qualys.com&ips=10.10.10.23&title=WEB_ORA_CREATE" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_weblogic/
```

### XML output:

```xml
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
```

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain=[value]</td>
<td>(Optional) A single Oracle WebLogic Server domain name. Example: website</td>
</tr>
<tr>
<td></td>
<td>The domain parameter must be specified with auto_discover=0 in the same request.</td>
</tr>
</tbody>
</table>

**Target Hosts**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips=[value]</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>(Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips=[value]</td>
<td>(Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips=[value]</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id=[value]</td>
<td>(Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>
<RESPONSE>
  <DATETIME>2018-03-10T13:30:49Z</DATETIME>
  <BATCH_LIST>
    <BATCH>
      <TEXT>Successfully Created</TEXT>
      <ID_SET>
        <ID>2707632279</ID>
      </ID_SET>
    </BATCH>
  </BATCH_LIST>
</RESPONSE>

Sample - Create WebLogic record, with auto discover

API request:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=create&installation_path=/u01/app/oracle&auto_discover=1&ips=10.10.10.23&title=ABC_ORA"
"https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_weblogic/
```

XML output:

```xml
<!DOCTYPE BATCH_RETURN SYSTEM
  "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-03-10T13:42:46Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>2707642279</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

DTDs for auth type “oracle_weblogic”

```
<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/oracle_weblogic/
auth_oracle_weblogic_list_output.dtd
```
Palo Alto Firewall Record

/api/2.0/fo/auth/palo_alto_firewall/

[POST]

Create, update, list and delete Palo Alto Firewall records for authenticated scans of Palo Alto Firewall instances. Vulnerability and compliance scans are supported (using VM, PC).

Requirements:
- The user account you provide for authentication must either have the predefined role “Superuser (read-only)” or a custom role with these XML API privileges enabled: Configuration and Operational Requests.
- We use the PANOS XML API to retrieve system information from Palo Alto Firewall on port 443 so this port must be open.

Tip - We strongly recommend you create one or more dedicated user accounts to be used solely by the Qualys Cloud Platform to authenticate to Palo Alto Firewall instances.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
</tbody>
</table>

Login credentials

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The username of the account to be used for authentication. If password is specified this is the username of a Palo Alto Firewall account. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>password={value}</td>
<td>(To create record password or login_type=vault is required) The password of the Palo Alto Firewall account to be used for authentication. Maximum 100 characters (ascii).</td>
</tr>
</tbody>
</table>
### Scan Authentication

#### Palo Alto Firewall Record

**Sample - Create Palo Alto Firewall record**

**API request:**

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d "action=create&title=palo-4&ips=10.10.10.10&login_type=basic&username=root&password=<PASSWORD>" "https://qualysapi.qualys.com/api/2.0/fo/auth/palo_alto_firewall/
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-01-14T06:29:41Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          login_type=vault (To create record password or login_type=vault is required)
          Set to vault if a third party vault will be used to retrieve password. Vault parameters need to be provided in the record.

### Target Hosts

#### ips={value}

- **Description:** (Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.

- **Description:** (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.

- **Description:** This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.

#### add_ips={value}

- **Description:** (Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated.

- **Description:** This parameter and the ips parameter cannot be specified in the same request.

#### remove_ips={value}

- **Description:** (Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.

- **Description:** This parameter and the ips parameter cannot be specified in the same request.
Scan Authentication

Palo Alto Firewall Record

Sample - Create Palo Alto Firewall record, using vault

API request:

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-01-16T06:22:01Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125726</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

Sample - List Palo Alto Firewall records

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d "action=list" "https://qualysapi.qualys.com/api/2.0/fo/auth/palo_alto_firewall/?action=list&ids=125727"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
"https://qualysapi.qualys.com/api/2.0/fo/auth/palo_alto_firewall/auth_palo_alto_firewall_list_output.dtd">

<AUTH_PALO_ALTO_FIREWALL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-09-13T06:30:32Z</DATETIME>
    <AUTH_PALO_ALTO_FIREWALL_LIST>
      <AUTH_PALO_ALTO_FIREWALL>
        <ID>125727</ID>
        <TITLE><![CDATA[palo-4]]></TITLE>
        <USERNAME><![CDATA[root]]></USERNAME>
        <SSL_VERIFY><![CDATA[1]]></SSL_VERIFY>
        <IP_SET>
          <IP>10.10.10.10</IP>
        </IP_SET>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <CREATED>
          <DATETIME>2017-09-13T06:29:41Z</DATETIME>
        </CREATED>
      </AUTH_PALO_ALTO_FIREWALL>
    </AUTH_PALO_ALTO_FIREWALL_LIST>
  </RESPONSE>
</AUTH_PALO_ALTO_FIREWALL_LIST_OUTPUT>

DTDs for auth type “palo_alto_firewall”

<platform API server>/api/2.0/batch_return.dtd

<platform API server>/api/2.0/fo/auth/palo_alto_firewall/auth_palo_alto_firewall_list_output.dtd
Pivotal Greenplum Record

/api/2.0/fo/auth/greenplum/

[POST]

List, create, update, and delete Pivotal Greenplum records for authenticated scans of Pivotal Greenplum 5.x and 6.x instances running on Unix. Compliance scans are supported (using PC).

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>details={value}</td>
<td>(Optional) Default value is Basic. You can choose from None, Basic, and All.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required only for update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>greenplum_unix_conf_file={value}</td>
<td>(Required for create request) The full path to the configuration file (postgresql.conf) on your Unix assets (IP addresses). The file must be in the same location on all assets for this record.</td>
</tr>
<tr>
<td>greenplum_db_name={value}</td>
<td>(Required for create request) The database instance you want to authenticate to.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Optional) The port where the database instance is running. Default is 5432.</td>
</tr>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
<tr>
<td>hosts={value}</td>
<td>(Required if ssl_verify=1) A list of FQDNs for all host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed.</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required for create request) The username of the account to be used for authentication. If password is specified this is the username of a Greenplum account. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).</td>
</tr>
</tbody>
</table>

Greenplum

Login credentials
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>password={value}</td>
<td>(For create request, password or login_type=vault is required) The password of the Greenplum account to be used for authentication. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td>login_type={value}</td>
<td>(For create request, password or login_type=vault is required) Login type can be basic (default) or vault. Set to vault if a third party vault will be used to retrieve the password. Vault parameters need to be provided in the record. See Vault Definition.</td>
</tr>
<tr>
<td>vault_id={value}</td>
<td>(Required if login_type=vault) The ID of the vault to be used to retrieve the password for login.</td>
</tr>
<tr>
<td>vault_type={value}</td>
<td>(Required if login_type=vault) The third party vault to be used to retrieve the password for login. Certain vaults support this capability. See Vault Support matrix.</td>
</tr>
<tr>
<td>client_key_type={value}</td>
<td>(Optional) Client key type basic (default) or vault.</td>
</tr>
<tr>
<td>client_key={value}</td>
<td>(Optional if client_key_type=basic) Client key content, if private key not in vault.</td>
</tr>
<tr>
<td>client_key_vault_type={value}</td>
<td>(Required if client_key_type=vault) The third party vault to be used to retrieve the private key. Certain vaults support this capability. See Vault Support matrix.</td>
</tr>
<tr>
<td>client_key_vault_id={value}</td>
<td>(Required if client_key_type=vault) The ID of the vault to get the private key from.</td>
</tr>
<tr>
<td>passphrase_type={value}</td>
<td>(Optional) Passphrase type can be basic (default) or vault.</td>
</tr>
<tr>
<td>passphrase={value}</td>
<td>(Optional if passphrase_type=basic) The passphrase value.</td>
</tr>
<tr>
<td>client_cert={value}</td>
<td>(Optional if passphrase_type=basic) The passphrase certificate content.</td>
</tr>
<tr>
<td>passphrase_vault_type={value}</td>
<td>(Required if passphrase_type=vault) The vault where the private key passphrase is stored. For example: CA Access Control, CyberArk AIM, Thycotic Secret Server.</td>
</tr>
<tr>
<td>passphrase_vault_id={value}</td>
<td>(Required if passphrase_type=vault) The ID of the vault to get the passphrase from.</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
</tbody>
</table>

### Keys, Passphrase

- **client_key_type={value}** (Optional) Client key type basic (default) or vault.
- **client_key={value}** (Optional if client_key_type=basic) Client key content, if private key not in vault.
- **client_key_vault_type={value}** (Required if client_key_type=vault) The third party vault to be used to retrieve the private key. Certain vaults support this capability. See Vault Support matrix.
- **client_key_vault_id={value}** (Required if client_key_type=vault) The ID of the vault to get the private key from.
- **passphrase_type={value}** (Optional) Passphrase type can be basic (default) or vault.
- **passphrase={value}** (Optional if passphrase_type=basic) The passphrase value.
- **client_cert={value}** (Optional if passphrase_type=basic) The passphrase certificate content.
- **passphrase_vault_type={value}** (Required if passphrase_type=vault) The vault where the private key passphrase is stored. For example: CA Access Control, CyberArk AIM, Thycotic Secret Server.
- **passphrase_vault_id={value}** (Required if passphrase_type=vault) The ID of the vault to get the passphrase from.

### Target Hosts

- **ips={value}** (Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.
Sample: List all record types

**API request:**
```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With:curl' -d "action=list" "https://qualysapi.qualys.com/api/2.0/fo/auth/
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_RECORDS_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/auth_records.dtd">
<AUTH_RECORDS_OUTPUT>
  <RESPONSE>
    <DATETIME>2019-10-04T09:24:19Z</DATETIME>
    <AUTH_RECORDS>
      <AUTH_UNIX_IDS>
        <ID_SET>
          <ID>1029116</ID>
          <ID>1296290</ID>
          <ID_RANGE>1375563-1375564</ID_RANGE>
          <ID>1505926</ID>
        </ID_SET>
      </AUTH_UNIX_IDS>
      <AUTH_GREENPLUM_IDS>
        <ID_SET>
          <ID>1505929</ID>
        </ID_SET>
      </AUTH_GREENPLUM_IDS>
    </AUTH_RECORDS>
  </RESPONSE>
</AUTH_RECORDS_OUTPUT>
```
Sample - List Greenplum Records with All Details

API request:
```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' -d "action=list&details=All"
"https://qualysapi.qualys.com/api/2.0/fo/auth/greenplum/"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_GREENPLUM_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/greenplum/auth_greenplum_list_output.dtd">
<AUTH_GREENPLUM_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-01-05T11:41:28Z</DATETIME>
    <AUTH_GREENPLUM_LIST>
      <AUTH_GREENPLUM>
        <ID>66186</ID>
        <TITLE>
          <![CDATA[greenplum auth]]>
        </TITLE>
        <USERNAME>
          <![CDATA[root]]>
        </USERNAME>
        <DATABASE>
          <![CDATA[postgres]]>
        </DATABASE>
        <PORT>5432</PORT>
        <SSL_VERIFY>
          <![CDATA[0]]>
        </SSL_VERIFY>
        <IP_SET>
          <IP>10.20.32.111</IP>
        </IP_SET>
        <UNIX_CONF_FILE>
          <![CDATA[/usr/local/greenplum-db/master/gpseg-1/postgresql.conf]]>
        </UNIX_CONF_FILE>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2019-12-31T10:51:10Z</DATETIME>
          <BY>qualys_jd</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2019-12-31T10:51:10Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_GREENPLUM>
    </AUTH_GREENPLUM_LIST>
  </RESPONSE>
</AUTH_GREENPLUM_LIST_OUTPUT>
```
Scan Authentication
Pivotal Greenplum Record

</AUTH_GREENPLUM>
<AUTH_GREENPLUM>
  <ID>66390</ID>
  <TITLE>
    <![CDATA[my greenplum record]]>
  </TITLE>
  <USERNAME>
    <![CDATA[root]]>
  </USERNAME>
  <DATABASE>
    <![CDATA[postgres]]>
  </DATABASE>
  <PORT>5432</PORT>
</AUTH_GREENPLUM>
<AUTH_GREENPLUM_LIST>
  <UNIX_CONF_FILE>
    <![CDATA[ /var/lib/pgsql/data/postgresql.conf]]>
  </UNIX_CONF_FILE>
</AUTH_GREENPLUM_LIST_OUTPUT>
Sample - Create Greenplum Record

API request:
```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' -d "action=create&title=my greenplum record&ips=10.10.10.1&username=root&password=<PASSWORD>&greenplum_db_name=postgres&port=5421&greenplum_unix_conf_path=/tmp/postgresql.conf"
"https://qualysapi.qualys.com/api/2.0/fo/auth/greenplum/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-01-05T12:04:32Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>66391</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Update Greenplum Record

API request:
```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' -d "action=update&ids=66391&title=my greenplum record&comments=new comment"
"https://qualysapi.qualys.com/api/2.0/fo/auth/greenplum/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-01-05T12:09:25Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Sample - Delete Greenplum Records

API request:


XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-01-05T12:10:16Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>66391</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

DTDs for auth type “greenplum”

<platform API server>/api/2.0/fo/auth/auth_records.dtd

<platform API server>/api/2.0/fo/auth/greenplum/auth_greenplum_list_output.dtd
**PostgreSQL Record**

/api/2.0/fo/auth/postgresql/

**[POST]**

Create, update, list and delete PostgreSQL records for authenticated scans of PostgreSQL 9.x, PostgreSQL 10.x, PostgreSQL 11.x and PostgreSQL 12.x instances running on Windows or Unix. Compliance scans are supported (using PC).

Requirement - You must configure login credentials on target hosts before scanning.

**Qualys User Guide - PostreSQL Authentication (.zip)**

Tip - We strongly recommend you create one or more dedicated user accounts to be used solely by the Qualys Cloud Platform to authenticate to PostgreSQL database instances.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>pgsql_win_conf_path={value}</td>
<td>(Optional) The full path to the PostgreSQL configuration file on your Windows assets (IP addresses). The file must be in the same location on all assets for this record.</td>
</tr>
<tr>
<td>pgsql_unix_conf_file={value}</td>
<td>(Optional) The full path to the PostgreSQL configuration file on your Unix assets (IP addresses). The file must be in the same location on all assets for this record.</td>
</tr>
<tr>
<td>pgsql_db_name={value}</td>
<td>(Required for create request) The database instance you want to authenticate to.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Optional) The port where the database instance is running. Default is 5432.</td>
</tr>
<tr>
<td>hosts={value}</td>
<td>(Required if ssl_verify=1) A list of FQDNs for all host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed.</td>
</tr>
</tbody>
</table>
### Parameter

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>**ssl_verify={0</td>
</tr>
</tbody>
</table>

### Login credentials

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>username={value}</strong> (Required for create request) The username of the account to be used for authentication. If password is specified this is the username of a PostgreSQL account. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td><strong>password={value}</strong> (For create request, password or login_type=vault is required) The password of the PostgreSQL account to be used for authentication. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td><strong>login_type=vault</strong> (To create record password or login_type=vault is required) Set to vault if a third party vault will be used to retrieve password. Vault parameters need to be provided in the record. See <a href="#">Vault Definition</a>.</td>
</tr>
</tbody>
</table>

### Keys, Passphrase

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>client_key_type={value}</strong> (Optional) Client key type basic (default) or vault.</td>
</tr>
<tr>
<td><strong>client_key={value}</strong> (Optional if client_key_type=basic) Client key content, if private key not in vault.</td>
</tr>
<tr>
<td><strong>client_key_vault_type={value}</strong> (Required if client_key_type=vault) The third party vault to be used to retrieve the private key. Certain vaults support this capability. See <a href="#">Vault Support matrix</a>.</td>
</tr>
<tr>
<td><strong>client_key_vault_id={value}</strong> (Required if client_key_type=vault) The ID of the vault to get the private key from.</td>
</tr>
<tr>
<td><strong>passphrase_type={value}</strong> (Optional) Passphrase type can be basic (default) or vault.</td>
</tr>
<tr>
<td><strong>passphrase={value}</strong> (Optional if passphrase_type=basic) The passphrase value.</td>
</tr>
<tr>
<td><strong>client_cert={value}</strong> (Optional if passphrase_type=basic) The passphrase certificate content.</td>
</tr>
<tr>
<td><strong>passphrase_vault_type={value}</strong> (Required if passphrase_type=vault) The vault where the private key passphrase is stored. For example CA Access Control, CyberArk AIM, Thycotic Secret Server.</td>
</tr>
<tr>
<td><strong>passphrase_vault_id={value}</strong> (Required if passphrase_type=vault) The ID of the vault to get the passphrase from.</td>
</tr>
</tbody>
</table>
Sample - Create PostgreSQL Record on Unix

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d "action=create&title=API_POSTGRE_2&username=root&password=<PASSWORD>&pgsql_db_name=presql&ips=10.10.10.35&pgsql_unix_conf_path=/etc&network_id=4002"
"https://qualysapi.qualys.com/api/2.0/fo/auth/postgresql/" > file.xml
```

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-03-27T20:17:42Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Target Hosts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>
Sample - Create PostgreSQL Record on Windows

API request:

curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' -d "action=create&title=api-windows-postgres&pgsql_win_conf_path=C:\Program Files\PostgreSQL\11\data\postgresql.conf&pgsql_db_name=postgres&username=qualys_scan&password=<<PASSWORD>>&ips=10.10.10.35"
"https://qualysapi.qualys.com/api/2.0/fo/auth/postgresql"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-01-28T10:55:39Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>72178</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

Sample - Update PostgreSQL Record on Unix

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d "action=update&ids=84307&add_ips=10.10.10.40-10.10.10.42"
"https://qualysapi.qualys.com/api/2.0/fo/auth/postgresql/" > file.xml

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
Sample - Update PostgreSQL Record on Windows

API request:

curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' -d "action=update&ids=72178&pgsql_win_conf_path=C:\Program Files\PostgreSQL\11\data\postgresql11.conf" "https://qualysapi.qualys.com/api/2.0/fo/auth/postgresql"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-01-28T11:06:36Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>72178</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

Sample - List PostgreSQL Records

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d "action=list&details=All" "https://qualysapi.qualys.com/api/2.0/fo/auth/postgresql/" > file.xml

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_POSTGRESQL_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/postgresql/auth_postgresql_list_output.dtd">
<AUTH_POSTGRESQL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-04-24T22:01:50Z</DATETIME>
    <AUTH_POSTGRESQL_LIST>
      <AUTH_POSTGRESQL>
        <ID>79518</ID>
        <TITLE><![CDATA[PostgesSQL1]]></TITLE>
        <USERNAME><![CDATA[acme_as1]]></USERNAME>
        <DATABASE><![CDATA[mydb1]]></DATABASE>
        <PORT>5432</PORT>
        <SSL_VERIFY><![CDATA[0]]></SSL_VERIFY>
        <IP_SET>
          <IP>10.10.10.45</IP>
        </IP_SET>
        <WIN_CONF_FILE><![CDATA[C:\Program Files\pgsql\data\postgresql.conf]]></WIN_CONF_FILE>
        <UNIX_CONF_FILE><![CDATA[/var/lib/pgsql/9.3/data/postgresql.conf]]></UNIX_CONF_FILE>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2018-04-13T23:42:50Z</DATETIME>
          <BY>acme_as1</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2018-04-20T23:35:42Z</DATETIME>
        </LAST_MODIFIED>
        <COMMENTS><![CDATA[my comments]]></COMMENTS>
      </AUTH_POSTGRESQL>
      <AUTH_POSTGRESQL>
        <ID>82110</ID>
        <TITLE><![CDATA[PostgreSQL2]]></TITLE>
        <USERNAME><![CDATA[acme_as1]]></USERNAME>
        <DATABASE><![CDATA[mydb2]]></DATABASE>
        <PORT>5432</PORT>
        <SSL_VERIFY><![CDATA[1]]></SSL_VERIFY>
        <HOSTS>
          <HOST><![CDATA[cent-31-107.ml2k8.qualys.com]]></HOST>
        </HOSTS>
        <IP_SET>
          <IP>10.20.31.107</IP>
        </IP_SET>
        <WIN_CONF_FILE><![CDATA[C:\Program Files\pgsql\data\postgresql.conf]]></WIN_CONF_FILE>
      </AUTH_POSTGRESQL>
    </AUTH_POSTGRESQL_LIST>
  </RESPONSE>
</AUTH_POSTGRESQL_LIST_OUTPUT>
Files\pgsql\data\postgresql.conf]></WIN_CONF_FILE>
<UNIX_CONF_FILE><![CDATA[/var/lib/pgsql/9.3/data/postgresql.conf]]>
</UNIX_CONF_FILE>
   <NETWORK_ID>0</NETWORK_ID>
   <CREATED>
      <DATETIME>2018-04-20T20:12:48Z</DATETIME>
      <BY>acme_as1</BY>
   </CREATED>
   ...
</AUTH_POSTGRESQL_LIST>
</RESPONSE>
</AUTH_POSTGRESQL_LIST_OUTPUT>

**DTDs for auth type “postgresql”**

<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/postgresql/auth_postgresql_list_output.dtd
SAP Hana Record
/api/2.0/fo/auth/sap_hana/

[POST]

SAP Hana authentication is supported for compliance scans (using PC or SCA). The SAP Hana API (api/2.0/fo/auth/sap_hana/) lets you list, create, update and delete SAP Hana authentication records. User permissions for this API are the same as other authentication record APIs.

Input Parameters
Use these parameters to create or update SAP Hana authentication records.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>database={value}</td>
<td>(Required for create request) The name of the database you want to authenticate to.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Required for create request) The port the database is on.</td>
</tr>
<tr>
<td>unix_conf_path={value}</td>
<td>(Required for create request when this record will be used for scanning Unix hosts) The SAP Hana configuration path on Unix hosts (up to 255 multi-byte characters).</td>
</tr>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
<tr>
<td>hosts={value}</td>
<td>(Required if ssl_verify=1) A list of FQDNs for all host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed. Multiple hosts are comma separated.</td>
</tr>
</tbody>
</table>

Login credentials
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username={value}</td>
<td>(Required for create request) The username of the account to be used for authentication. If password is specified this is the username of a SAP Hana account. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>password={value}</td>
<td>(For create request, password or login_type=vault is required) The password of the SAP Hana account to be used for authentication. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td>password_encryption={0</td>
<td>1}</td>
</tr>
<tr>
<td>login_type={value}</td>
<td>(For create request, password or login_type=vault is required) Login type can be basic (default) or vault. Set to vault if a third party vault will be used to retrieve the password. Vault parameters need to be provided in the record. See Vault Definition.</td>
</tr>
<tr>
<td>vault_id={value}</td>
<td>(Required if login_type=vault) The ID of the vault to be used to retrieve the password for login.</td>
</tr>
<tr>
<td>vault_type={value}</td>
<td>(Required if login_type=vault) The third party vault to be used to retrieve the password for login. Certain vaults support this capability. See Vault Support matrix.</td>
</tr>
<tr>
<td>Target Hosts</td>
<td></td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) for the targets you want to authenticate to. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>(Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional to create or update record, and valid only when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>
Sample - Create SAP Hana Record

**API request:**
```
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
    <RESPONSE>
        <DATETIME>2021-01-12T14:39:46Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Created</TEXT>
                <ID_SET>
                    <ID>4474043</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```

Sample - Update SAP Hana Record

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=update&ids=4474043&comments=update1" "https://qualysapi.qualys.com/api/2.0/fo/auth/sap_hana/
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
    <RESPONSE>
        <DATETIME>2021-01-12T14:45:58Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Updated</TEXT>
                <ID_SET>
                    <ID>4474043</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```
Sample - List SAP Hana Records with All Details

**API request:**
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=list&details=All" "https://qualysapi.qualys.com/api/2.0/fo/auth/sap_hana/
```

**XML output:**
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_SAP_HANA_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/sap_hana/auth_sap_hana_list_output.dtd">
<AUTH_SAP_HANA_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-01-12T14:34:42Z</DATETIME>
    <AUTH_SAP_HANA_LIST>
      <AUTH_SAP_HANA>
        <ID>4474042</ID>
        <TITLE><![CDATA[SAP_HANA]]></TITLE>
        <USERNAME><![CDATA[SYSTEM]]></USERNAME>
        <DATABASE><![CDATA[SYSTEMDB]]></DATABASE>
        <PORT>39013</PORT>
        <SSL_VERIFY><![CDATA[1]]></SSL_VERIFY>
        <HOSTS>
          <HOST><![CDATA[host.domain1]]></HOST>
        </HOSTS>
        <IP_SET>
          <IP>10.11.70.185</IP>
        </IP_SET>
        <UNIX_CONF_PATH><![CDATA[/etc/saphana.conf]]></UNIX_CONF_PATH>
        <PASSWORD_ENCRYPTION><![CDATA[1]]></PASSWORD_ENCRYPTION>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <CREATED>
          <DATETIME>2021-01-12T14:28:16Z</DATETIME>
          <BY>joe_user</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2021-01-12T14:33:05Z</DATETIME>
        </LAST_MODIFIED>
        <COMMENTS><![CDATA[created successfully]]></COMMENTS>
      </AUTH_SAP_HANA>
    </AUTH_SAP_HANA_LIST>
    <GLOSSARY>
      <USER_LIST>
        <USER>
          <USER_LOGIN>joe_user</USER_LOGIN>
          <FIRST_NAME>Joe</FIRST_NAME>
          <LAST_NAME>User</LAST_NAME>
        </USER>
      </USER_LIST>
    </GLOSSARY>
  </RESPONSE>
</AUTH_SAP_HANA_LIST_OUTPUT>
```
Sample - Delete SAP Hana Records

API request:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=delete&ids=4474043"
"https://qualysapi.qualys.com/api/2.0/fo/auth/sap_hana/
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-01-12T14:48:56Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>4474043</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

DTDs for auth type “sap hana”

```xml
<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/sap_hana/auth_sap_hana_list_output.dtd
```
Scan Authentication
SAP IQ Record

/api/2.0/fo/auth/sapiq/

[POST]

The SAP IQ API lets you list, create, update and delete SAP IQ authentication records for compliance scans (using PC). User permissions for this API are the same as other authentication record APIs.

Input Parameters
Use these parameters to create or update SAP IQ authentication records.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>database={value}</td>
<td>(Required for create request) The name of the database you want to authenticate to.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Required for create request) The port the database is running on.</td>
</tr>
<tr>
<td>installation_dir={value}</td>
<td>(Required for create request when this record will be used for scanning Unix hosts) The database installation directory for scanning Unix hosts.</td>
</tr>
<tr>
<td>login_type=vault</td>
<td>(Required for create request) The username of the account to be used for authentication. If password is specified this is the username of a SAP IQ account. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>password={value}</td>
<td>(For create request, password or login_type=vault is required) The password of the SAP IQ account to be used for authentication. Maximum 100 characters (ascii).</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>password_encryption=[0</td>
<td>1]</td>
</tr>
<tr>
<td>login_type=[value]</td>
<td>(For create request, password or login_type=vault is required) Login type can be basic (default) or vault. Set to vault if a third party vault will be used to retrieve the password. Vault parameters need to be provided in the record. See Vault Definition.</td>
</tr>
<tr>
<td>vault_id=[value]</td>
<td>(Required if login_type=vault) The ID of the vault to be used to retrieve the password for login.</td>
</tr>
<tr>
<td>vault_type=[value]</td>
<td>(Required if login_type=vault) The third party vault to be used to retrieve the password for login. Certain vaults support this capability. See Vault Support matrix.</td>
</tr>
<tr>
<td>ips=[value]</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips=[value]</td>
<td>(Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips=[value]</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id=[value]</td>
<td>(Optional to create or update record, and valid only when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>

### Target Hosts

- **ips=[value]** (Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.
- (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.
- This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.

- **add_ips=[value]** (Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated.
- This parameter and the ips parameter cannot be specified in the same request.

- **remove_ips=[value]** (Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.
- This parameter and the ips parameter cannot be specified in the same request.

Sample - Create SAP IQ Record

**API request:**

```
```
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-12-05T12:04:32Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>96171</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Update SAP IQ Record

API request:

```bash
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-12-11T10:47:46Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>4423386</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - List SAP IQ Records with All Details

API request:

```bash
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' -d "action=list&details=All" "https://qualysapi.qualys.com/api/2.0/fo/auth/sapiq/
```
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_SAPIQ_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/sapiq/auth_sapiq_list_output.dtd">
<AUTH_SAPIQ_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-12-11T18:02:56Z</DATETIME>
    <AUTH_SAPIQ_LIST>
      <AUTH_SAP_IQ>
        <ID>4423387</ID>
        <TITLE><![CDATA[sap_iq_api_2]]></TITLE>
        <USERNAME><![CDATA[dba]]></USERNAME>
        <IP_SET>
          <IP>10.11.70.54</IP>
        </IP_SET>
        <DATABASE><![CDATA[iqdemo]]></DATABASE>
        <PORT>2638</PORT>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2020-12-11T06:24:15Z</DATETIME>
          <BY>joe_user</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2020-12-11T06:24:15Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_SAP_IQ>
      <AUTH_SAP_IQ>
        <ID>4423518</ID>
        <TITLE><![CDATA[sap_iq_api_3]]></TITLE>
        <USERNAME><![CDATA[dba]]></USERNAME>
        <IP_SET>
          <IP>10.11.70.52</IP>
        </IP_SET>
        <DATABASE><![CDATA[iqdemo]]></DATABASE>
        <PORT>2638</PORT>
        <INSTALLATION_DIR><![CDATA[test]]></INSTALLATION_DIR>
        <PASSWORD_ENCRYPTION><![CDATA[1]]></PASSWORD_ENCRYPTION>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2020-12-11T12:35:12Z</DATETIME>
          <BY>joe_user</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2020-12-11T12:35:12Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_SAP_IQ>
    </AUTH_SAPIQ_LIST>
  </RESPONSE>
</AUTH_SAPIQ_LIST_OUTPUT>
<GLOSSARY>
  <USER_LIST>
    <USER>
      
    </USER>
  </USER_LIST>
</GLOSSARY>
```

Sample - Delete SAP IQ Records

API request:

curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' -d "action=delete&ids=4423386" "https://qualysapi.qualys.com/api/2.0/fo/auth/sapiq/"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-12-11T10:53:04Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>4423386</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

DTDs for auth type “sap iq”

<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/sapiq/auth_sapiq_list_output.dtd
SNMP Record

/api/2.0/fo/auth/snmp/

[POST]

Create, update, list and delete SNMP records for authenticated scans of SNMP enabled devices. Supported are vulnerability and compliance scans (using VM, PC). Supported versions are SNMPv1, SNMPv2 and SNMPv3.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>version={v1</td>
<td>v2c</td>
</tr>
</tbody>
</table>

Login credentials

| community_strings={value} | (Optional and valid using SNMPv1 and SNMPv2c) The SNMP community strings to be used for authentication to target hosts. Multiple entries are comma separated. The service attempts authentication using several common default community strings. When community_strings is specified, the user-provided community strings are used for authentication before the default community strings. |
**Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username={value}</td>
<td>(Optional and valid using SNMPv3) The user account for authentication to target hosts. A maximum of 128 characters may be specified.</td>
</tr>
<tr>
<td></td>
<td>These three parameters are used to specify authentication: username, password and auth_alg.</td>
</tr>
<tr>
<td></td>
<td>If creating a record and authentication will be used, it is required that all three parameters are specified together. If updating a record to change the username, the username specified will replace the existing username in the record. If updating a record to remove authentication, specify an empty value for all three parameters.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Optional and valid using SNMPv3) The password for authentication to target hosts. Maximum of 128 characters.</td>
</tr>
<tr>
<td></td>
<td>These three parameters are used to specify authentication: username, password and auth_alg.</td>
</tr>
<tr>
<td></td>
<td>If creating a record and authentication will be used, it is required that all three parameters are specified together. If updating a record to change the password, the password specified will replace the existing password in the record. If updating a record to remove authentication, specify an empty value for all three parameters.</td>
</tr>
<tr>
<td>auth_alg={MD5</td>
<td>SHA1}</td>
</tr>
<tr>
<td></td>
<td>These three parameters are used to specify authentication: username, password and auth_alg.</td>
</tr>
<tr>
<td></td>
<td>If creating a record and authentication will be used, it is required that all three parameters are specified together. If updating a record to change the authentication algorithm, the algorithm specified will replace the existing algorithm in the record. If updating a record to remove authentication, specify an empty value for all three parameters.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>encrypt_password={value}</code></td>
<td>(Optional and valid using SNMPv3) The password if privacy (data encryption) is to be used for SNMP communication. Maximum of 128 characters. These two parameters are used to specify privacy: <code>encrypt_password</code> and <code>priv_alg</code>. If creating a record and privacy will be used, it is required that both parameters are specified together. If updating a record to change the password, the password specified will replace the existing password in the record. If updating a record to remove privacy, specify an empty value for both parameters.</td>
</tr>
<tr>
<td>`priv_alg={DES</td>
<td>AES}`</td>
</tr>
<tr>
<td><code>security_engine_id={value}</code></td>
<td>(Optional and valid using SNMPv3) The security engine ID when a security engine is part of the target host configuration. A valid ID is required. A maximum of 128 characters may be specified. If a security engine ID is part of the target host configuration, the parameter <code>security_engine_id</code> must be defined for the record in order for authentication to be successful. If the security engine ID is not defined (and is required by the target host for all SNMP requests), then the SNMP service may not be detected on the target host and authentication will fail.</td>
</tr>
<tr>
<td><code>context_engine_id={value}</code></td>
<td>(Optional and valid using SNMPv3) The context engine ID used in scoped PDUs when a context is part of the target host configuration. A valid ID is required. A maximum of 128 characters may be specified. If an SNMP context is part of the target host configuration, the parameters <code>context_engine_id</code> and/or <code>context</code> must be defined for the record in order for the scanning engine to retrieve context-sensitive information from the target host.</td>
</tr>
</tbody>
</table>
Scan Authentication
SNMP Record

Sample - Create SNMP record, using SNMPv3

API request:

curl -H "X-Requested-With: Cutil Sample" -d "action=create&title=My+Record&version=v3&username=user&password=<PASSWORD>&auth_alg=MD5&encrypt_password=<PASSWORD>&priv_alg=DES&security_engine_id=0x80001F88805131F121BD9B194B&context_engine_id=0x80001F88805131F121BD9B194B&context=bridge1&ips=10.10.10.2-10.10.10.4" -b "QualysSession=a3863e31b486417f81eea7f8881f3142; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/auth/snmp/"

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>context={value}</td>
<td>(Optional and valid using SNMPv3) The context name used in scoped PDUs when a context is part of the target host configuration. A maximum of 128 characters may be specified. If an SNMP context is part of the target host configuration, the parameters context_engine_id and/or context must be defined for the record in order for the scanning engine to retrieve context-sensitive information from the target host.</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-02-27T06:22:01Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125726</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Update an SNMP record

Change the user name and password for authentication and the target IPs.

```bash
curl -H "X-Requested-With: Curl Sample" -d "action=update&ids=65319&username=user2&password=<PASSWORD>&ips=10.10.10.5-10.10.10.6" -b "QualysSession=a3863e31b486417f81eeaf881f3142; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/auth/snmp/
```

DTDs for auth type “snmp”

- `<platform API server>/api/2.0/batch_return.dtd`
- `<platform API server>/api/2.0/fo/auth/snmp/auth_snmp_list_output.dtd`
Sybase Record

/api/2.0/fo/auth/sybase/

[POST]

Create, update, list and delete Sybase records for authenticating to Sybase Adaptive Server Enterprise (ASE) instances. Sybase auth records are supported for VM & PC.

Requirement - You must configure login credentials on target hosts before scanning.

Download Qualys User Guide - Sybase Authentication (.zip)

Tip - We strongly recommend you create one or more dedicated user accounts to be used solely by the Qualys Cloud Platform to authenticate to Sybase database instances.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Required to create record) The port the Sybase database is on.</td>
</tr>
<tr>
<td>database={value}</td>
<td>(Optional to create and update record) The name of the Sybase database you want to authenticate to.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_discover_databases=[0</td>
<td>1]</td>
</tr>
<tr>
<td>installation_dir={value}</td>
<td>(Required for create request if this record will be used for scanning Unix hosts) The database installation directory for scanning Unix hosts.</td>
</tr>
</tbody>
</table>

### Login credentials

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The username of the account to be used for authentication. If password is specified this is the username of a Sybase account. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>password={value}</td>
<td>(To create record password or login_type=vault is required) The password of the Sybase account to be used for authentication. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td>password_encryption={0</td>
<td>1}</td>
</tr>
<tr>
<td>login_type=vault</td>
<td>(To create record password or login_type=vault is required) Set to vault if a third party vault will be used to retrieve password. Vault parameters need to be provided in the record. See Vault Definition</td>
</tr>
</tbody>
</table>

### Target Hosts
Sample - Create Sybase Record

API request:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d "action=create&title=sybase_record&network_id=19015&username=acme_ac12&password=<PASSWORD>&port=444&database=sybaseDB1&ips=10.10.24.12,10.10.24.13,10.10.24.15&installation_dir=/dir123&comments=This%20Sybase%20comments" "https://qualysapi.qualys.com/api/2.0/fo/auth/sybase/" > file.xml
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-10T20:52:31Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>78782</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Sample - Create Sybase Record, with vault
API request:

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
 <RESPONSE>
  <DATETIME>2018-04-18T18:54:36Z</DATETIME>
  <BATCH_LIST>
   <BATCH>
    <TEXT>Successfully Created</TEXT>
    <ID_SET>
     <ID>88888</ID>
    </ID_SET>
   </BATCH>
  </BATCH_LIST>
 </RESPONSE>
</BATCH_RETURN>

Sample - Create Sybase Record to enable password encryption and auto discovery
API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d "action=create&title=sybase_record&network_id=19015&username=acme_ac12&password=<PASSWORD>&password_encryption=1&auto_discover_databases=1&port=444&installation_dir=/dir123&comments=This%20Sybase%20comments" "https://qualysapi.qualys.com/api/2.0/fo/auth/sybase/" > file.xml
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "http://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-04-18T15:45:05Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>43025</ID>
          <ID>43025</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Update Sybase Record

**API request:**

```bash
```

Sample - List Sybase records

**API request:**

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d "action=list&details=All" "https://qualysapi.qualys.com/api/2.0/fo/auth/sybase/" > file.xml
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_SYBASE_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/sybase/auth_sybase_list_output.dtd">
<AUTH_SYBASE_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-04-10T21:32:21Z</DATETIME>
    <AUTH_SYBASE_LIST>
      <AUTH_SYBASE>
        <ID>78177</ID>
        <TITLE><![CDATA[api_syb_basic_2IPs_NW2]]></TITLE>
      </AUTH_SYBASE>
    </AUTH_SYBASE_LIST>
  </RESPONSE>
</AUTH_SYBASE_LIST_OUTPUT>
```
<USERNAME><![CDATA[api_user1]]></USERNAME>
<DATABASE><![CDATA[api_sybDB1]]></DATABASE>
<PORT>444</PORT>
</IP_SET>
  <IP_RANGE>10.10.24.12-10.10.24.13</IP_RANGE>
</IP_SET>
<NETWORK_ID>19019</NETWORK_ID>
<CREATED>
  <DATETIME>2017-04-08T00:17:17Z</DATETIME>
  <BY>enter_ss</BY>
</CREATED>
<LAST_MODIFIED>
  <DATETIME>2017-04-08T00:17:17Z</DATETIME>
</LAST_MODIFIED>
</AUTH_SYBASE>

<AUTH_SYBASE>
  <ID>78186</ID>
  <TITLE><![CDATA[api_syb_basic_2IPs_Global]]></TITLE>
  <USERNAME><![CDATA[api_user1]]></USERNAME>
  <DATABASE><![CDATA[api_sybDB1]]></DATABASE>
  <PORT>444</PORT>
  <IP_SET>
    <IP_RANGE>10.10.24.12-10.10.24.13</IP_RANGE>
  </IP_SET>
  <NETWORK_ID>0</NETWORK_ID>
  <CREATED>
    <DATETIME>2017-04-08T01:10:04Z</DATETIME>
    <BY>enter_ss</BY>
  </CREATED>
  <LAST_MODIFIED>
    <DATETIME>2017-04-08T01:10:04Z</DATETIME>
  </LAST_MODIFIED>
</AUTH_SYBASE>

...
**Unix Record**

/api/2.0/fo/auth/unix/

[POST]

Create, update, list and delete Unix records for authenticated scans of hosts running on Unix, Cisco and Checkpoint Firewall. Vulnerability and compliance scans are supported on Unix and Cisco systems (using VM, PC). Compliance scans are supported on Checkpoint Firewall systems (using PC).

**Download Qualys User Guide - Unix Authentication (pdf)**

**Input Parameters**

Parameters: Request | Login credentials| Unix only | Target Hosts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>sub_type={cisco</td>
<td>checkpoint_firewall}</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Optional and valid for compliance scans only) Custom ports to be used to perform authenticated compliance assessment (control testing).</td>
</tr>
</tbody>
</table>

**Login credentials**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The username of the account to be used for authentication. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>password={value}</td>
<td>(To create record password or login_type=vault is required) The password of the PostgreSQL account to be used for authentication when a vault will not be used. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>login_type={basic</td>
<td>vault}</td>
</tr>
</tbody>
</table>
| cleartext_password={0|1}                | (Optional) When not specified, the scanning engine only uses strong password encryption for remote login. Specify 1 to allow your password to be transmitted in clear text when connecting to services which do not support strong password encryption. For more info, search for “Clear Text Password” in online help. For a create request, if cleartext_password=1, the password parameter is required. For an update request, if cleartext_password=1, and the record does not have a password set, then cleartext_password=1 is *silently ignored*.

| skip_password={0|1}                  | (Optional and valid only for Unix record, i.e not supported for Cisco or Checkpoint Firewall sub-type) By default when only the required parameters are set (title, username, ips) the login account password is set to the empty password. You can set skip_password=1 if the login account does not have a password. When set it’s not possible to set the empty password, another password using the “password” parameter, or password in a vault. |
| enable_password={value}             | (Optional and valid only for Cisco sub-type) The password required for executing the “enable” command on the target hosts. Maximum 100 characters (ascii). Note: The pooled credentials feature is not supported if the “enable” command requires a password and it is specified using the enable_password parameter. |
| expert_password={value}             | (Optional and valid only for Checkpoint Firewall sub-type) The password required for executing the “expert” command on the target hosts. Maximum 100 characters (ascii).                                                                                                                                                                                                                                                  |
| target_type={value}                 | (Optional) Specify the target type. You can choose from the following values:
- A10
- HP_COMWARE
- CISCO_ASA_WITH_FIREPOWE
- auto (default)                                                                                                                                                                                                                                                                                                    |

**Kerberos/GSSAPI authentication details, if it is enabled for the target host**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>use_kerberos={0</td>
<td>1}</td>
</tr>
<tr>
<td>realm_discovery={value}</td>
<td>(Mandatory, if 'use_kerberos=1') Specify the realm discovery method. The available values are manual, single, and DNS.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_realm={value}</td>
<td>(Mandatory, if 'use_kerberos=1') Specify the name of the realm that a user belongs to.</td>
</tr>
<tr>
<td>service_realm={value}</td>
<td>(Mandatory, if 'use_kerberos=1') Specify the name of the realm that the service belongs to, when a user wants to access a service that is part of a different realm. <strong>Note:</strong> This parameter is valid only if the “realm_discovery” parameter is set to “manual”.</td>
</tr>
<tr>
<td>service_kdc={value}</td>
<td>(Optional) Specify the KDC that manages authentication for the service in its realm, when a user wants to access a service that is part of a different realm. <strong>Note:</strong> This parameter is valid only if the “realm_discovery” parameter is set to “manual”.</td>
</tr>
<tr>
<td>user_kdc={value}</td>
<td>(Optional) Specify the KDC (Key Distribution Center) that is responsible for authenticating users and issuing ticket-granting tickets (TGTs) for the realm.</td>
</tr>
<tr>
<td>krb5_password={value}</td>
<td>(Mandatory, if 'use_kerberos=1') Enter the password to authenticate to the Kerberos Key Distribution Center (KDC).</td>
</tr>
<tr>
<td>krb5_login_type={value}</td>
<td>(Optional) Specify the type of login used to authenticate to the Kerberos Key Distribution Center (KDC). The available values are “basic” and “vault”.</td>
</tr>
<tr>
<td>krb5_{&lt;vaultparameters&gt;={value}}</td>
<td>(Mandatory, if krb5_login_type = vault) If krb5_login_type is 'vault', then all vault parameter fields must be added with the prefix 'krb5_'. For example, krb5_vault_type, krb5_vault_id, etc. The vault-specific parameters depend on the vault type you have selected. See the “Vault Definition” section in the API user guide.</td>
</tr>
<tr>
<td>[XML File]</td>
<td>(Optional and valid only for Unix record, i.e. not supported for Cisco or Checkpoint Firewall sub-type) XML file where you define private-key certificates and root delegations. These are defined using this DTD: &lt;platform API server&gt;/api/2.0/fo/auth/unix/unix_auth_params.dtd</td>
</tr>
<tr>
<td>use_agentless_tracking={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
### Parameter

<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>agentless_tracking_path={value}</td>
<td>(Required if use_agentless_tracking=1 for Unix record, i.e. not supported for Cisco or Checkpoint Firewall sub-type) The pathname where you would like the service to store the host ID file on each host. This is required to enable Agentless Tracking for Unix.</td>
</tr>
</tbody>
</table>

### Target Hosts

<table>
<thead>
<tr>
<th><strong>ips={value}</strong></th>
<th>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>
### Target Hosts with Tag Support

Note: Applicable only when you have Asset Tagging and Tag Support for Authentication Records enabled for your subscription.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| asset_type={ips|asset_tags|ip_range_tag_rule} | (Optional) Indicates how assets will be defined in the record. Valid values are ips (the default), asset_tags, ip_range_tag_rule. When not specified, we'll use asset_type=ips.  
  - ips - Specify this value to assign IP addresses/ranges to the record.  
  - asset_tags - Specify this value to add tags to the record for the assets you want included. IP addresses with the selected tags already assigned will be associated with the record.  
  - ip_range_tag_rule - Specify this value to add tags that have IP address ranges defined in the tag rule. All IP addresses defined in the tag rule will be associated with the record, including IPs that don't already have the tag assigned. |
| tag_set_by={id|name} | (Optional when asset_type=asset_tags or ip_range_tag_rule) Specify “id” (the default) to select a tag set by providing tag IDs. Specify “name” to select a tag set by providing tag names. |
| tags_include={tag1,tag2...} | (Required when asset_type=asset_tags or ip_range_tag_rule) Specify a tag set to include in the record. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. To specify tag names, you must also specify tag_set_by=name. |
| tags_exclude={tag1,tag2,...} | (Optional when asset_type=asset_tags or ip_range_tag_rule) Specify a tag set to exclude from the record. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. To specify tag names, you must also specify tag_set_by=name. |
| tag_include_selector={any|all} | (Optional when asset_type=asset_tags or ip_range_tag_rule) Select “any” (the default) to include hosts that match at least one of the selected tags. Select “all” to include hosts that match all of the selected tags. |
| tag_exclude_selector={any|all} | (Optional when asset_type=asset_tags or ip_range_tag_rule) Select “any” (the default) to exclude hosts that match at least one of the selected tags. Select “all” to exclude hosts that match all of the selected tags. |
Scan Authentication
Unix Record

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record when asset_type=ips or asset_type is not specified) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated. (Optional to update record when asset_type=ips) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record when asset_type=ips) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record when asset_type=ips) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated. This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
</tbody>
</table>

**Ports Used For Unix Compliance Scans**
The actual ports used for compliance scanning (Unix, Cisco, Checkpoint Firewall) depends on scan settings in 1) compliance option profile, and 2) Unix authentication record as indicated.

<table>
<thead>
<tr>
<th>Compliance Option Profile</th>
<th>Authentication Record</th>
<th>Ports Scanned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Scan</td>
<td>UI; Well Known Ports</td>
<td>~ 1900 Ports (includes Ports 22, 23, 513)</td>
</tr>
<tr>
<td></td>
<td>API: no “port” parameter</td>
<td></td>
</tr>
<tr>
<td>Standard Scan</td>
<td>UI: Custom Ports</td>
<td>~ 1900 Ports + Custom Ports in record</td>
</tr>
<tr>
<td></td>
<td>API: “port” parameter</td>
<td></td>
</tr>
<tr>
<td>Targeted Scan</td>
<td>UI: Well Known Ports</td>
<td>Ports 22, 23 and 513 only</td>
</tr>
<tr>
<td></td>
<td>API: no “port” parameter</td>
<td></td>
</tr>
<tr>
<td>Targeted Scan</td>
<td>UI: Custom Ports</td>
<td>Custom Ports in record</td>
</tr>
<tr>
<td></td>
<td>API: “port” parameter</td>
<td></td>
</tr>
</tbody>
</table>
Sample - Create Unix record, with password
Applies to record type Unix, Cisco and Checkpoint Firewall

API request:
```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=create&title=Unix&username=root&password=<PASSWORD>}&ips=10.10.36.63"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-18T18:54:36Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>12345</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Create Unix record, root delegation tools and vault
Applies to record type Unix only (not sub-types)

API request:
```
curl -H "X-Requested-With: curl" -H "Content-type:text/xml" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=create&title=Unix&vault&username=Qualys&ips=10.113.195.152&login_type=vault&system_name=0&auto_discover System=0&custom_system_type=custom" --data-binary @add_params.xml

add_params.xml
```
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<UNIX_AUTH_PARAMS>
  <ROOT_TOOLS>
    <ROOT_TOOL>
      <STANDARD_TYPE type="pimsu"/>
    </ROOT_TOOL>
  </ROOT_TOOLS>
</UNIX_AUTH_PARAMS>
```

420
Scan Authentication
Unix Record

<ROOT_TOOL>
  <CUSTOM_TYPE><![CDATA[test]]></CUSTOM_TYPE>
  <PASSWORD_INFO type="basic">
    <PASSWORD><![CDATA[password]]></PASSWORD>
  </PASSWORD_INFO>
</ROOT_TOOL>

<PRIVATE_KEY_CERTIFICATES>
  <PRIVATE_KEY_CERTIFICATE>
    <PRIVATE_KEY_INFO type="vault">
      <DIGITAL_VAULT>
        <VAULT_TYPE>CyberArk AIM</VAULT_TYPE>
        <VAULT_ID>25026922</VAULT_ID>
        <FOLDER><![CDATA[folder]]></FOLDER>
        <FILE><![CDATA[file]]></FILE>
      </DIGITAL_VAULT>
    </PRIVATE_KEY_INFO>
    <PASSPHRASE_INFO type="basic">
      <PASSPHRASE><![CDATA[passphrase]]></PASSPHRASE>
    </PASSPHRASE_INFO>
  </PRIVATE_KEY_CERTIFICATE>
  <PRIVATE_KEY_CERTIFICATE>
    <PRIVATE_KEY_INFO type="basic">
      <PRIVATE_KEY type="rsa">
        <![CDATA[-----BEGIN RSA PRIVATE KEY-----
       Proc-Type: 4,ENCRYPTED
       DEK-Info: AES-128-CBC,F9A653E2D12E019357B349B6EEE068B1
        F1fGHOc0rREmC0cBPslvqvaitPNTGeqKRMssSwGNaZNTAcKs1soY/WkMDW6QD
        dLZNiGB0CFag94zyoMyCyrdpayACAOWfH5wBVixxHFL6Vxx5b6foLBE40FOYAIP
        sdm1HvCf9FNa2dFf10unb0erwjiqJNwY1V78529e1E+2+dzIemi90ibh0R35NB60
        TLeS3UUvezp/09ZPLfOqPPhNgfW4Gxp/SUwjoES9fCQE+BW4MMWHWu8XKtytt
        ....
        -----END RSA PRIVATE KEY-----]]></PRIVATE_KEY>
    </PRIVATE_KEY_INFO>
    <PASSPHRASE_INFO type="vault">
      <DIGITAL_VAULT>
    </PASSPHRASE_INFO>
  </PRIVATE_KEY_CERTIFICATE>
</PRIVATE_KEY_CERTIFICATES>
Scan Authentication
Unix Record

<VAULT_USERNAME><![CDATA[PASSPHRASE USERNAME]]></VAULT_USERNAME>
<VAULT_TYPE>Quest Vault</VAULT_TYPE>
<VAULT_ID>35046922</VAULT_ID>

<SYSTEM_NAME><![CDATA[quest_system_name]]></SYSTEM_NAME>

</DIGITAL_VAULT>
</PASSPHRASE_INFO>

<CERTIFICATE type="openssh">
<![CDATA[ssh-rsa-cert-v01@openssh.com AAAAHHNzaC1yc2EytYVydC12MDFAb3B1bnNzaC5jb20AAAAGwR4bJ5iBT10gCAQuf 3y261o2WyfnB1OEsQ45Kb9qLgAAAAAQABAAABQAQC5sVLb7emh8/v2uHp6x1pN5R+M Hqwt3A5M/GkKtuui1Jnc/XYgqeWLMOJpbVtCVXwUcgKt4Q0D1qgc4uhZhrZdtpQG HrEivndNNLYV9NQj7LozE7x/sGiWdtmlucHu1teXMaBmp4aER9Y6u5v6y17CAY9 bcVZ/1j1SypmjzkPj39aJq+Qx2kIv+H4uh/T05LvHdi1FrjWWwE018DV/DRIw3h8o 4jhjn1QxBxyjad3efmFaejgRnY6cBw821gm...]]></CERTIFICATE>

</PRIVATE_KEY_CERTIFICATE>
</PRIVATE_KEYCERTIFICATES>
</UNIX_AUTH_PARAMS>

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-18T18:54:36Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>....</ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
Sample - Create Unix Record with target type set to HP_COMWARE

Provide a target type while creating or updating the Unix (SSH2) authentication record.

**API request:**
```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=create&
title=ux-target-type&username=root&ips=10.11.42.114&login_type=basic&password=<PASSWORD>&target_type=HP_COMWARE
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-05-26T21:17:17Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>149016</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Create Unix Record with Tags

In this sample, a new Unix record is created with asset_type=ip_range_tag_rule.

**API request:**
```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=create&
title=unix&username=root&asset_type=ip_range_tag_rule&tags_include=7515612&tag_include_selector=all&tags_exclude=7514462&tag_exclude_selector=all"
```
Sample - Create unix auth record with Kerberos authentication details

**API request:**

curl --location --request POST
'https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?username=root&
anction=create&ips=10.0.0.1&title=unix
krbsapi&use_kerberos=1&realm_discovery=manual&user_realm=realm.com
&service_realm=abc.com&service_kdc=kdc&user_kdc=kerbs&
krb5_password=<PASSWORD>&krb5_login_type=basic'
--header 'X-Requested-With: portal'
--header 'Authorization: Basic <token>'

**API response**

```xml
< xml version="1.0" encoding="UTF-8" >
< !DOCTYPE AUTH_UNIX_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/dtd/auth_list_output.dtd">
< AUTH_UNIX_LIST_OUTPUT >
< RESPONSE >
< DATETIME >2023-02-13T04:08:26Z</ DATETIME >
< AUTH_UNIX_LIST >
< AUTH_UNIX >
< ID >214497</ ID >
< TITLE >
<![CDATA[krbs]]>
< /TITLE >
</ AUTH_UNIX >
</ AUTH_UNIX_LIST >
</ RESPONSE >
</ AUTH_UNIX_LIST_OUTPUT >
```
<USERNAME>
  <![CDATA[root]]>
</USERNAME>

<SKIP_PASSWORD>0</SKIP_PASSWORD>

<CLEARTEXT_PASSWORD>0</CLEARTEXT_PASSWORD>

<TARGET_TYPE>
  <![CDATA[Auto (default)]]>
</TARGET_TYPE>

<KERBEROS_AUTHENTICATION>1</KERBEROS_AUTHENTICATION>

<REALM_DISCOVERY>
  <![CDATA[manual]]>
</REALM_DISCOVERY>

<USER_REALM>
  <![CDATA[jsm.com]]>
</USER_REALM>

<KERBEROS_AUTHENTICATION>
  <![CDATA[Auto (default)]]>
</KERBEROS_AUTHENTICATION>

<REALM_DISCOVERY>
  <![CDATA[manual]]>
</REALM_DISCOVERY>

<USER_KDC>
  <![CDATA[kerbs.jsm.com]]>
</USER_KDC>

<SERVICE_REALM>
  <![CDATA[kerbs.jsm.com]]>
</SERVICE_REALM>

<SERVICE_KDC>
  <![CDATA[krb]]>
</SERVICE_KDC>

<IP_SET>
  <IP>0.0.0.0</IP>
</IP_SET>

<NETWORK_ID>0</NETWORK_ID>

<CREATED>
  <DATETIME>2023-02-06T09:48:20Z</DATETIME>
  <BY>test_pq4</BY>
</CREATED>

<LAST_MODIFIED>
  <DATETIME>2023-02-06T12:30:33Z</DATETIME>
</LAST_MODIFIED>

</AUTH_UNIX>

<AUTH_UNIX>
  <ID>214498</ID>
  <TITLE>
    <![CDATA[k1]]>
  </TITLE>
  <USERNAME>
    <![CDATA[root]]>
  </USERNAME>
</AUTH_UNIX>

<SKIP_PASSWORD>0</SKIP_PASSWORD>
<CLEARTEXT_PASSWORD>0</CLEARTEXT_PASSWORD>
<TARGET_TYPE>
  <![CDATA[Auto (default)]]>
</TARGET_TYPE>

<KERBEROS_AUTHENTICATION>1</KERBEROS_AUTHENTICATION>
<REALM_DISCOVERY>
  <![CDATA[manual]]>
</REALM_DISCOVERY>
<User_REALM>
  <![CDATA[ffwwqw]]>
</User_REALM>
<User_KDC>
  <![CDATA[user]]>
</User_KDC>
<Service_REALM>
  <![CDATA[sisdd]]>
</Service_REALM>
<Service_KDC>
  <![CDATA[]]>
</Service_KDC>
<KERBEROS_LOGIN_INFO type="vault">
  <DIGITAL_VAULT>
    <DIGITAL_VAULT_ID>
      <![CDATA[55014]]>
    </DIGITAL_VAULT_ID>
    <DIGITAL_VAULT_TYPE>
      <![CDATA[Quest Vault]]>
    </DIGITAL_VAULT_TYPE>
    <DIGITAL_VAULT_TITLE>
      <![CDATA[quest]]>
    </DIGITAL_VAULT_TITLE>
    <VAULT_SYSTEM_NAME>
      <![CDATA[fhk]]>
    </VAULT_SYSTEM_NAME>
  </DIGITAL_VAULT>
</KERBEROS_LOGIN_INFO>
<IP_SET>
  <IP>0.0.0.0</IP>
</IP_SET>
<N networks_id>0</NETWORK_ID>
<CREATED>
  <DATETIME>2023-02-06T12:54:00Z</DATETIME>
  <BY>test_pq4</BY>
</CREATED>
<LAST_MODIFIED>
Sample - Update Unix auth record with target type CISCO_ASA_WITH_FIREPOWE

API request:
```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=update&ids=149016&target_type=CISCO_ASA_WITH_FIREPOWE
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-05-26T21:34:18Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>149016</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - List Unix auth record with to view updated target type

API request:
```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=list&ids=149016
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_UNIX_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/auth_unix_list_output.dtd">
<AUTH_UNIX_LIST_OUTPUT>
  <RESPONSE>
```

427
<DATETIME>2020-05-26T21:35:23Z</DATETIME>
<AUTH_UNIX_LIST>
  <AUTH_UNIX>
    <ID>149016</ID>
    <TITLE>
      <![CDATA[ux-target-type]]>
    </TITLE>
    <USERNAME>
      <![CDATA[root]]>
    </USERNAME>
    <TARGET_TYPE>
      <![CDATA[Cisco Adaptive Security Appliance with FirePower]]>
    </TARGET_TYPE>
    <IP_SET>
      <IP>10.11.42.114</IP>
    </IP_SET>
    <NETWORK_ID>0</NETWORK_ID>
    <CREATED>
      <DATETIME>2020-05-26T21:17:17Z</DATETIME>
      <BY>username</BY>
    </CREATED>
    <LAST_MODIFIED>
      <DATETIME>2020-05-26T21:34:18Z</DATETIME>
    </LAST_MODIFIED>
  </AUTH_UNIX>
</AUTH_UNIX_LIST>

Sample - List Unix auth records

API request:
```
url -k -u agms_nb:Qatemp123# -s -S -H 'X-Requested-With:curl demo2' -d "action=list" "qualys_base_url/api/2.0/fo/auth/unix/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_UNIX_LIST_OUTPUT SYSTEM "qualys_base_url/api/2.0/fo/auth/unix/dtd/auth_list_output.dtd">
<AUTH_UNIX_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2023-09-05T08:48:12Z</DATETIME>
    <AUTH_UNIX_LIST>
      <AUTH_UNIX>
```
<ID>321277</ID>
<TITLE><![CDATA[API_v2_END_TO_END_agms_nb_UNIX]]></TITLE>
<USERNAME><![CDATA[root]]></USERNAME>
<SKIP_PASSWORD>0</SKIP_PASSWORD>
<CLEARTEXT_PASSWORD>0</CLEARTEXT_PASSWORD>
/IP_SET>
    <IP>10.10.10.10</IP>
</IP_SET>
<NETWORK_ID>0</NETWORK_ID>
<CREATED>
    <DATETIME>2019-10-15T08:48:06Z</DATETIME>
    <BY>agms_nb</BY>
</CREATED>
<LAST_MODIFIED>
    <DATETIME>2019-10-15T08:48:06Z</DATETIME>
</LAST_MODIFIED>
<QUALYS_SHELL>
    <ENABLED>0</ENABLED>
</QUALYS_SHELL>
</AUTH_UNIX>

...

<AUTH_UNIX>
<ID>3382464</ID>
<TITLE><![CDATA[unn]]></TITLE>
<USERNAME><![CDATA[admin]]></USERNAME>
<SKIP_PASSWORD>0</SKIP_PASSWORD>
<CLEARTEXT_PASSWORD>0</CLEARTEXT_PASSWORD>
<TARGET_TYPE><![CDATA[Auto (default)]]></TARGET_TYPE>
<PRIVATE_KEY_CERTIFICATE_LIST>
    <PRIVATE_KEY_CERTIFICATE>
        <ID>2257361</ID>
        <PRIVATE_KEY_INFO type="basic">
            <PRIVATE_KEY type="pkcs8" />
        </PRIVATE_KEY_INFO>
        <PASSPHRASE_INFO type="basic" />
    </PRIVATE_KEY_CERTIFICATE>
    <PRIVATE_KEY_CERTIFICATE>
        <ID>2257360</ID>
        <PRIVATE_KEY_INFO type="basic">
            <PRIVATE_KEY type="pkcs8" />
        </PRIVATE_KEY_INFO>
        <PASSPHRASE_INFO type="basic" />
    </PRIVATE_KEY_CERTIFICATE>
    <PRIVATE_KEY_CERTIFICATE>
        <ID>2255896</ID>
        <PRIVATE_KEY_INFO type="basic">
            <PRIVATE_KEY type="pkcs8" />
        </PRIVATE_KEY_INFO>
        <PASSPHRASE_INFO type="basic" />
    </PRIVATE_KEY_CERTIFICATE>
</PRIVATE_KEY_CERTIFICATE_LIST>
<PRIVATE_KEY_INFO type="basic">
  <PRIVATE_KEY type="pkcs8" />
</PRIVATE_KEY_INFO>

<PRIVATE_KEY_CERTIFICATE>
  <PRIVATE_KEY_CERTIFICATE_LIST> <IP_SET> <IP>1.9.0.8</IP> </IP_SET> <NETWORK_ID>0</NETWORK_ID> <CREATED>
    <DATETIME>2023-08-23T09:40:11Z</DATETIME>
    <BY>agms_nb</BY>
  </CREATED> <LAST_MODIFIED>
    <DATETIME>2023-08-24T10:04:07Z</DATETIME>
  </LAST_MODIFIED> <QUALYS_SHELL>
    <ENABLED>0</ENABLED>
  </QUALYS_SHELL>
</PRIVATE_KEY_CERTIFICATE>

More Samples
Qualys API - Unix Authentication API samples (GitHub)

DTDs for auth type “unix”
<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/unix/auth_unix_list_output.dtd

For Unix type record type only, root delegation tools and private-key certificates are specified using the unix_auth_params.dtd here
<platform API server>/api/2.0/fo/auth/unix/unix_auth_params.dtd
Network SSH Record

/api/2.0/fo/auth/network_ssh/

[POST]

Network SSH authentication is supported for vulnerability and compliance scans. The new Network SSH API (/api/2.0/fo/auth/network_ssh/) lets you list, create, update and delete Network SSH authentication records. This authentication supports SSH2 format.

Network SSH authentication record can be used in place of the Cisco and Checkpoint Firewall authentication records. This authentication record has all the same functionality as the Cisco and Checkpoint Firewall records and additional support for target_type field similar to Unix authentication record.

Network SSH authentication records support for password and password2 fields with vaults. This password2 field is similar to expert_password field (for Checkpoint Firewall sub-type) and enable_password field (for Cisco sub-type).

### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Optional) The port the database name is running on.</td>
</tr>
<tr>
<td>target_type={value}</td>
<td>(Optional) Specify the target type.</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required for create request) The username of the account to be used for authentication. If password is specified this is the username of a Network SSH account. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Optional) The password of the Network SSH account to be used for authentication. Maximum 100 characters (ascii).</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cleartext_password=[0</td>
<td>1]</td>
</tr>
<tr>
<td>password2={value}</td>
<td>(Optional) This password2 field is similar to existing expert_password field (for Checkpoint Firewall sub-type) and enable_password field (for Cisco sub-type). For Checkpoint Firewall: The password required for executing the “expert” command on the target hosts. The password may include 1-31 characters (ascii). For Cisco: The password required for executing the “enable” command on the target hosts. The password may include 1-31 characters (ascii).</td>
</tr>
<tr>
<td>login_type={value}</td>
<td>(Optional) Login type can be basic (default) or vault. Set to vault if a third party vault will be used to retrieve the password. Vault parameters need to be provided in the record. See “Vault Definition” in the API user guide.</td>
</tr>
<tr>
<td>vault_id={value}</td>
<td>(Required if login_type=vault) The ID of the vault to be used to retrieve the password for login.</td>
</tr>
<tr>
<td>vault_type={value}</td>
<td>(Required if login_type=vault) The third party vault to be used to retrieve the password for login. Certain vaults support this capability. See “Vault Support Matrix” in the API user guide.</td>
</tr>
<tr>
<td>p2_login_type={value}</td>
<td>(Optional) p2 Login type can be basic (default) or vault. Set to vault if a third party vault will be used to retrieve the password. Vault parameters need to be provided in the record. See “Vault Definition” in the API user guide.</td>
</tr>
<tr>
<td>p2_&lt;vault parameters&gt;={value}</td>
<td>(Optional) If p2_login_type is vault then all vault parameter fields must be added with prefix ‘p2_‘. For example, p2_vault_type, p2_vault_id. Vault specific parameters required depend on the vault type you’ve selected. See “Vault Definition” in the API user guide.</td>
</tr>
</tbody>
</table>
Scan Authentication
Network SSH Record

Sample - Create Network SSH Authentication Record

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"https://qualysapi.qualys.com/api/2.0/fo/auth/network_ssh/?action=
create&username=abc&title=a11&ips=10.10.110.12&password=<PASSWORD>
&port=270,17,122&cleartext_password=1&target_type=A10&password2=12
34"

API request using xml file:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"https://qualysapi.qualys.com/api/2.0/fo/auth/network_ssh/?action=
create&username=abc&title=new%201&ips=10.10.110.12&password=<PASSWORD>
&comments=new%20auth%20record&port=270,17,122&cleartext_password=
1&target_type=A10&p2_login_type=vault&p2_vault_type=Thycotic%20
Secret%20Server&p2_vault_id=41014&p2_secret_name=sc_name&password2
=1234&login_type=vault&vault_type=Thycotic%20Secret%20Server&vault
_id=41014&secret_name=bder&details=All"
--data-binary @add_params.xml

Content of add_params.xml

<?xml version="1.0" encoding="UTF-8" ?>
<NETWORK_SSH_AUTH_PARAMS>
PRIVATE_KEY_CERTIFICATES
PRIVATE_KEY_CERTIFICATE
PRIVATE_KEY_INFO type="vault"
DIGITAL_VAULT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) for the targets you want to authenticate to. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>(Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td></td>
<td>An IP added to the Network SSH authentication record cannot added in Unix, Cisco or Checkpoint authentication records.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>[XML File]</td>
<td>(Optional) XML file where you define private-key certificates. These are defined using this DTD: &lt;platform API server&gt;/api/2.0/fo/auth/network_ssh/network_ssh_auth_params.dtd</td>
</tr>
</tbody>
</table>
<VAULT_TYPE>CA_PAM</VAULT_TYPE>
<Vault_ID>41022</Vault_ID>
<Vault_DEVICE_NAME>hq_device</Vault_DEVICE_NAME>
<Vault_APP_NAME>APP_NAME</Vault_APP_NAME>
</DIGITAL_VAULT>
</PRIVATE_KEY_INFO>
</PASSPHRASE_INFO type="vault">
</DIGITAL_VAULT>
</VAULT_TYPE>CA_PAM</VAULT_TYPE>
<Vault_ID>41022</Vault_ID>
<Vault_DEVICE_NAME>hq_device</Vault_DEVICE_NAME>
<Vault_APP_NAME>APP_NAME</Vault_APP_NAME>
</DIGITAL_VAULT>
</PASSPHRASE_INFO>
</PRIVATE_KEY_CERTIFICATE>
</PRIVATE_KEY_CERTIFICATES>
</NETWORK_SSH_AUTH_PARAMS>

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"http://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-04-21T06:34:05Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>102451</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

Sample - Update Network SSH Authentication Record

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"https://qualysapi.qualys.com/api/2.0/fo/auth/network_ssh/?username=abc&password2=1234&action=update&ids=102419"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
Sample - Delete Network SSH Records

**API request:**
```
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-01-12T14:48:56Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>4474043</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - List Network SSH Records

**API request:**
```
curl -k -u agms_nb:Qatemp123# -s -S -H 'X-Requested-With:curl demo2' -d "action=list" "<qualys_base_url>/api/2.0/fo/auth/network_ssh/
```

435
Scan Authentication
Network SSH Record

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_NETWORK_SSH_LIST_OUTPUT SYSTEM "<qualys_base_url>/api/2.0/fo/auth/network_ssh/dtd/auth_list_output.dtd">
<br/>&lt;AUTH_NETWORK_SSH_LIST_OUTPUT&gt;
<br/>&lt;RESPONSE&gt;
<br/>&lt;DATETIME&gt;2023-09-05T08:44:22Z&lt;/DATETIME&gt;
<br/>&lt;AUTH_NETWORK_SSH_LIST&gt;
<br/>&lt;AUTH_NETWORK_SSH&gt;
<br/>&lt;ID&gt;3441133&lt;/ID&gt;
<br/>&lt;TITLE&gt;&lt;![CDATA/API_Unix_Auth_Unix_21]]&gt;&lt;/TITLE&gt;
<br/>&lt;USERNAME&gt;&lt;![CDATA[Qualys]]&gt;&lt;/USERNAME&gt;
<br/>&lt;IP_SET&gt;
<br/>&lt;IP&gt;10.10.10.10&lt;/IP&gt;
<br/>&lt;/IP_SET&gt;
<br/>&lt;CLEARTEXT_PASSWORD&gt;0&lt;/CLEARTEXT_PASSWORD&gt;
<br/>&lt;TARGET_TYPE&gt;&lt;![CDATA[Auto (default)]]&gt;&lt;/TARGET_TYPE&gt;
<br/>&lt;PRIVATE_KEY_CERTIFICATE_LIST&gt;
<br/>&lt;PRIVATE_KEY_CERTIFICATE&gt;
<br/>&lt;ID&gt;2274448&lt;/ID&gt;
<br/>&lt;PRIVATE_KEY_INFO type="basic">
<br/>&lt;PRIVATE_KEY type="pkcs8" /
<br/>&lt;/PRIVATE_KEY_INFO>
<br/>&lt;PASSPHRASE_INFO type="basic" /&gt;
<br/>&lt;/PRIVATE_KEY_CERTIFICATE&gt;
<br/>&lt;/PRIVATE_KEY_CERTIFICATE_LIST&gt;
<br/>&lt;NETWORK_ID&gt;0&lt;/NETWORK_ID&gt;
<br/>&lt;CREATED&gt;
<br/>&lt;DATETIME&gt;2023-09-04T15:43:07Z&lt;/DATETIME&gt;
<br/>&lt;BY&gt;agms_nb&lt;/BY&gt;
<br/>&lt;/CREATED&gt;
<br/>&lt;LAST_MODIFIED&gt;
<br/>&lt;DATETIME&gt;2023-09-04T15:43:07Z&lt;/DATETIME&gt;
<br/>&lt;/LAST_MODIFIED&gt;
<br/>&lt;/AUTH_NETWORK_SSH&gt;
<br/>&lt;/AUTH_NETWORK_SSH_LIST&gt;
<br/>&lt;/RESPONSE&gt;
<br/>&lt;/AUTH_NETWORK_SSH_LIST_OUTPUT&gt;
```

**DTDs for auth type “network_ssh”**

- `<platform API server>/api/2.0/batch_return.dtd`
- `<platform API server>/api/2.0/fo/auth/network_ssh/dtd/auth_list_output.dtd`
Private-key certificates are specified using the network_ssh_auth_params.dtd here
<platform API server>/api/2.0/fo/auth/network_ssh/network_ssh_auth_params.dtd
VMware Record

/api/2.0/fo/auth/vmware/

[POST]

Create, update, list and delete VMware records for authenticating to vSphere components running vSphere v4.x and 5.x. Vulnerability and compliance scans are supported (using VM, PC).

How it works - The VMware record allows for connections to the vSphere API for vSphere 5.x and 4.x. The vSphere API is a SOAP API used by all vSphere components, including VMware ESXi, VMware ESX, VMware vCenter Server, and the VMware vCenter Server Appliance. By default, the API connection occurs over an encrypted SSL web services connection on port 443.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
</tbody>
</table>

Login credentials

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username={value}</td>
<td>(Required to create record, optional to update record) The user name for a VMware account. A maximum of 13 characters (ascii) may be specified.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(To create record password or login_type=vault is required) The password for a VMware account. Maximum 100 characters (ascii).</td>
</tr>
<tr>
<td>login_type={}</td>
<td>(To create record password or login_type=vault is required) Set to vault if a third party vault will be used to retrieve password. Vault parameters need to be provided in the record. See Vault Definition. Set to “vcenter” to scan ESXi hosts through vCenter. The VMware record will include your ESXi IP addresses. You also need a vCenter authentication record with the vCenter IP addresses that map to your ESXi hosts.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Optional) The service communicates with ESXi web services on port 443 and another port can be configured. When unspecified, port 443 is used.</td>
</tr>
<tr>
<td>hosts={value}</td>
<td>(Optional) A list of FQDNs for the hosts that correspond to all ESXi host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed. Multiple hosts are comma separated.</td>
</tr>
<tr>
<td>ssl_verify={value}</td>
<td>(Optional) Specify “all” for a complete SSL certificate validation. Specify “skip” if the host SSL certificate is self-signed or uses an SSL certificate signed by a custom root CA. Specify “none” for no SSL verification.</td>
</tr>
<tr>
<td>is_disconnect={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

**Target Hosts**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>(Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If you set “is_disconnect=1” and add IPs that are already associated with a Unix record, the VMware ESXi record is not created or updated. Instead, an error is returned in the response. You must remove the IPs from the non-applicable record to resolve the error.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips={value}</td>
<td>(Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
</tbody>
</table>
Sample - Create VMware record

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=create&title=NewVMwareRecordWithAPI&username=USERNAME&password=PASSWORD&ips=10.10.10.2-10.10.10.4" "https://qualysapi.qualys.com/api/2.0/fo/auth/vmware/" > apiOutputCreateVMwareRecord.txt
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">  
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-02-13T21:16:41Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>30486</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Update VMware record

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=update&ids=1344232&is_disconnect=1" "https://qualysapi.qualys.com/api/2.0/fo/auth/vmware/"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">  
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-11-03T12:19:41Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>30486</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Sample - List VMware record

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X "POST" -d "action=list&details=All" "https://qualysapi.qualys.com/api/2.0/fo/auth/vmware/"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_VMWARE_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/vmware/auth_vmware_list_output.dtd">
<AUTH_VMWARE_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-11-22T07:32:21Z</DATETIME>
    <AUTH_VMWARE_LIST>
      <AUTH_VMWARE>
        <ID>409187</ID>
        <TITLE><![CDATA[VMware_Basic]]></TITLE>
        <USERNAME><![CDATA[root]]></USERNAME>
        <PORT>443</PORT>
        <SSL_VERIFY><![CDATA[skip]]></SSL_VERIFY>
        <IP_SET>
          <IP>10.20.30.40</IP>
        </IP_SET>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2020-01-23T07:55:13Z</DATETIME>
          <BY>joe_user</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2020-01-23T07:55:13Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_VMWARE>
      <AUTH_VMWARE>
        <ID>1344231</ID>
        <TITLE><![CDATA[VMware_Disconnected_Disabled]]></TITLE>
        <PORT>443</PORT>
      </AUTH_VMWARE>
    </AUTH_VMWARE_LIST>
  </RESPONSE>
</AUTH_VMWARE_LIST_OUTPUT>
```
<IP_SET>
  <IP>10.11.12.13</IP>
</IP_SET>

<LOGIN_TYPE><![CDATA[vcenter]]></LOGIN_TYPE>
<DISCONNECTED_ESXI>0</DISCONNECTED_ESXI>
<NETWORK_ID>0</NETWORK_ID>
<CREATED>
  <DATETIME>2021-11-03T12:09:53Z</DATETIME>
  <BY>joe_user</BY>
</CREATED>
<LAST_MODIFIED>
  <DATETIME>2021-11-10T13:11:23Z</DATETIME>
</LAST_MODIFIED>

</AUTH_VMWARE>

<AUTH_VMWARE>
  <ID>1344232</ID>
  <TITLE><![CDATA[VMware_Disconnected_Enabled]]></TITLE>
  <PORT>443</PORT>
  <IP_SET>
    <IP>8.9.10.11</IP>
  </IP_SET>
</AUTH_VMWARE>

<AUTH_VMWARE>
  <ID>1344232</ID>
  <TITLE><![CDATA[VMware_Disconnected_Enabled]]></TITLE>
  <PORT>443</PORT>
  <IP_SET>
    <IP>8.9.10.11</IP>
  </IP_SET>
</AUTH_VMWARE>

</AUTH_VMWARE_LIST>

<GLOSSARY>
  <USER_LIST>
    <USER>
      <USER_LOGIN>joe_user</USER_LOGIN>
      <FIRST_NAME>Joe</FIRST_NAME>
      <LAST_NAME>User</LAST_NAME>
    </USER>
  </USER_LIST>
</GLOSSARY>

</RESPONSE>

</AUTH_VMWARE_LIST_OUTPUT>
DTDs for auth type “vmware”
<platform API server>/api/2.0/batch_return.dtd
<platform API server>/api/2.0/fo/auth/vmware/auth_vmware_list_output.dtd
**Windows Record**

/\api/2.0/fo/auth/windows/

[POST]

Create, update, list and delete Windows records for authenticating to Windows systems. Vulnerability and Compliance scans are supported (using VM, PC).

Download Qualys User Guide - Windows Authentication (.pdf)

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record IDs to update/delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>use_agentless_tracking={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

**Login credentials**

| username={value} | (Required to create record, optional to update record) The username for the Windows account to be used for authentication on target hosts. The username may include 1-31 characters (ascii). |
| password={value} | (To create record password or login_type=vault is required) The password of the Windows account to be used for authentication. Maximum 100 characters (ascii). |
| login_type={basic|vault} | (To create record password or login_type=vault is required) Set to vault if a third party vault will be used to retrieve password. Vault parameters need to be provided in the record. See Vault Definition |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>windows_ad_domain={value}</td>
<td>(Optional) The Windows Active Directory domain name for domain level authentication. When specified, we’ll use an Active Directory forest to authenticate to hosts in a certain domain within the framework. You’ll need to enter a Fully Qualified Domain Name (FQDN). See Windows Domains. This parameter and the windows_domain parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>windows_domain={value}</td>
<td>(Optional) The Windows NetBIOS domain name for domain level authentication. See Windows Domains.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the windows_ad_domain parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td></td>
<td>When the ips parameter is also specified, the domain type is NetBIOS, User-Selected IPs. We’ll use NetBIOS to authenticate to the IPs in the domain configuration.</td>
</tr>
<tr>
<td></td>
<td>When the ips parameter is not specified, the domain type is NetBIOS, Service-Selected IPs. We’ll use NetBIOS to authenticate to hosts in the domain using credentials stored on the domain.</td>
</tr>
<tr>
<td>ntlm=[0</td>
<td>1]</td>
</tr>
</tbody>
</table>

### Target Hosts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>(Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
</tbody>
</table>
### Parameter |
### Description |
| add_ips={value} | (Optional to update record) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated. This parameter and the ips parameter cannot be specified in the same request. |
| remove_ips={value} | (Optional to update record) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated. This parameter and the ips parameter cannot be specified in the same request. |
| network_id={value} | (Optional to create or update record, and valid when the networks feature is enabled) The network ID for the record. |

#### Target Hosts with Tag Support

Note: Applicable only when you have Asset Tagging and Tag Support for Authentication Records enabled for your subscription.

### asset_type={ips|asset_tags|ip_range_tag_rule} | (Optional) Indicates how assets will be defined in the record. Valid values are ips (the default), asset_tags, ip_range_tag_rule. When not specified, we'll use asset_type=ips |
| ips - Specify this value to assign IP addresses/ranges to the record. |
| asset_tags - Specify this value to add tags to the record for the assets you want included. IP addresses with the selected tags already assigned will be associated with the record. |
| ip_range_tag_rule - Specify this value to add tags that have IP address ranges defined in the tag rule. All IP addresses defined in the tag rule will be associated with the record, including IPs that don’t already have the tag assigned. |

### tag_set_by={id|name} | (Optional when asset_type=asset_tags or ip_range_tag_rule) Specify “id” (the default) to select a tag set by providing tag IDs. Specify “name” to select a tag set by providing tag names. |

### tags_include={tag1,tag2...} | (Required when asset_type=asset_tags or ip_range_tag_rule) Specify a tag set to include in the record. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. To specify tag names, you must also specify tag_set_by=name. |
### Scan Authentication

**Windows Record**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tags_exclude={tag1,tag2,...} (Optional when asset_type=asset_tags or ip_range_tag_rule)</td>
<td>Specify a tag set to exclude from the record. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. To specify tag names, you must also specify tag_set_by=name.</td>
</tr>
<tr>
<td>tag_include_selector={any</td>
<td>all} (Optional when asset_type=asset_tags or ip_range_tag_rule)</td>
</tr>
<tr>
<td>tag_exclude_selector={any</td>
<td>all} (Optional when asset_type=asset_tags or ip_range_tag_rule)</td>
</tr>
<tr>
<td>ips={value} (Required to create record when asset_type=ips or asset_type is not specified)</td>
<td>The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>(Optional to update record when asset_type=ips) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>add_ips={value} (Optional to update record when asset_type=ips)</td>
<td>Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
<tr>
<td>remove_ips={value} (Optional to update record when asset_type=ips)</td>
<td>IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>This parameter and the ips parameter cannot be specified in the same request.</td>
</tr>
</tbody>
</table>
For Windows domain level authentication, all three authentication protocols are
supported. Kerberos and NTLMv2 are enabled by default in new records. If NTLM was enabled in
a record prior to this release, then NTLMv1 is enabled.

For Windows local host level authentication, NTLMv2 and NTLMv1 protocols are
supported. NTLMv2 is enabled by default in new records. If NTLM was enabled in a record prior
to this release, then NTLMv1 is enabled.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Protocols</strong></td>
<td></td>
</tr>
<tr>
<td>kerberos=0</td>
<td>1] (Optional) When not specified, Kerberos is enabled allowing the scanning engine to try Kerberos when negotiating authentication to target hosts. Specify kerberos=0 if you do not want Kerberos attempted. Kerberos is supported for domain authentication only. When kerberos=1 you must define a domain name for Windows Active Directory (windows_ad_domain) or NetBIOS (windows_domain) for the record.</td>
</tr>
<tr>
<td>ntlmv2=0</td>
<td>1] (Optional) When not specified for a new record, NTLMv2 is enabled allowing the scanning engine to try NTLMv2 when negotiating authentication to target hosts. Specify ntlmv2=0 if you do not want NTLMv2 attempted.</td>
</tr>
<tr>
<td>ntlm=0</td>
<td>1] (Optional) When not specified, NTLMv1 will not be attempted. Specify ntlm=1 to allow the scanning engine to try NTLMv1 when negotiating authentication to target hosts.</td>
</tr>
<tr>
<td><strong>SMB signing</strong></td>
<td></td>
</tr>
<tr>
<td>require_smb_signing=0</td>
<td>1] (Optional) Set to 0 (default) when SMB signing is not required. Set value to 1 to require SMB signing. Should I require SMB signing? The answer is No in most cases. If you enable this option in your record, we will require each Windows target to support SMB signing. If SMB signing is disabled on a target host, authentication will fail and the host will not be scanned. This option protects against MITM attacks but we won’t be able to authenticate to some hosts.</td>
</tr>
<tr>
<td>minimum_smb_version= {value} (Optional) The minimum SMB protocol version. Valid values are: 1, 2.0.2, 2.1, 3.0, 3.0.2, 3.1.1, and &quot;&quot; (empty string means no version set).</td>
<td></td>
</tr>
</tbody>
</table>
Windows Domains
- Supported domain types: Active Directory, NetBIOS User-Selected IPs, NetBIOS Service-Selected IPs.
- Authentication is performed at the local host level when a domain name is not defined for Active Directory (windows_ad_domain) or NetBIOS (windows_domain).
- Once a Windows record is saved, you cannot change the domain type from Active Directory to NetBIOS or from NetBIOS to Active Directory.

Sample - Create Windows Record

API request:
```bash
```

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/windows/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>30486</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Create Windows Record with Tags

In this sample, a new Windows record is created with asset_type=asset_tags.

API request:
```bash
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/api/2.0/fo/auth/windows/?action=create&title=windows&username=root&asset_type=asset_tags&tags_include=ag1&tag_include_selector=all&tags_exclude=ag20&tag_set_by=name&tag_exclude_selector=all"
```
Scan Authentication
Windows Record

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2021-03-11T00:45:31Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>204027</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - List windows records

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"action=list&ids=1310338&details=All"
"https://qualysapi.qualys.com/api/2.0/fo/auth/windows/
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_WINDOWS_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/windows/auth_windows
_list_output.dtd">
<AUTH_WINDOWS_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-04-30T09:29:45Z</DATETIME>
    <AUTH_WINDOWS_LIST>
      <AUTH_WINDOWS>
        <ID>1310338</ID>
        <TITLE><![CDATA[Windows_Record_1]]></TITLE>
        <USERNAME><![CDATA[acme_jd]]></USERNAME>
        <IP_SET>
          <IP>10.10.10.202</IP>
        </IP_SET>
        <CREATED>
          <DATETIME>2018-04-30T09:28:00Z</DATETIME>
        </CREATED>
      </AUTH_WINDOWS>
    </AUTH_WINDOWS_LIST>
  </RESPONSE>
</AUTH_WINDOWS_LIST_OUTPUT>
```
<BY>acme_jd</BY>
</CREATED>
<LAST_MODIFIED>
  <DATETIME>2018-04-30T09:28:43Z</DATETIME>
</LAST_MODIFIED>
<COMMENTS><![CDATA[My comments on Windows Record 1]]></COMMENTS>
</AUTH_WINDOWS>
</AUTH_WINDOWS_LIST>
<GLOSSARY>
  <USER_LIST>
    <USER>
      <USER_LOGIN>acme_jd</USER_LOGIN>
      <FIRST_NAME>John</FIRST_NAME>
      <LAST_NAME>Doe</LAST_NAME>
    </USER>
  </USER_LIST>
</GLOSSARY>
</RESPONSE>
</AUTH_WINDOWS_LIST_OUTPUT>

**DTDs for auth type “windows”**

- `<platform API server>/api/2.0/batch_return.dtd`
- `<platform API server>/api/2.0/fo/auth/windows/auth_windows_list_output.dtd`
Oracle HTTP Server Record

/api/2.0/fo/auth/oracle_http_server/

[POST]

Create, update, list and delete Oracle HTTP Server records for authenticating to Unix and Windows systems. Vulnerability and Compliance scans are supported (using VM, PC). User permissions for this API are the same as other authentication record APIs. Note that the API supports authentication record creation only for Oracle Server installed on respective OS - Unix or Windows.

Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title={value}</td>
<td>(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (asci).</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional and valid when the networks feature is enabled) The network ID for the record.</td>
</tr>
<tr>
<td>add_ips={value}</td>
<td>(Optional to update record) Add IPs to the IPs list for this record. Multiple IPs/ranges are comma separated.</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional to create or update record) User defined comments. Maximum of 1999 characters.</td>
</tr>
<tr>
<td>action={action}</td>
<td>(Required) Specify create, update, delete (using POST) or list (using GET or POST).</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required to create record) The IP address(es) the server will log into using the record’s credentials. Multiple entries are comma separated. (Optional to update record) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Required to update or delete record) Record Oracle HTTP type auth record IDs to update. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.</td>
</tr>
</tbody>
</table>

Unix Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unix_home_path={value}</td>
<td>(Required to create or update record if Unix working mode is selected) The root directory path for Oracle HTTP Server. Maximum of 255 characters.</td>
</tr>
<tr>
<td>unix_domain_path={value}</td>
<td>(Required to create or update record if Unix working mode is selected for Oracle HTTP Server 12c and higher) Absolute path to the top level directory where domains are configured. Maximum of 255 characters.</td>
</tr>
<tr>
<td>unix_inst_path={value}</td>
<td>(Required to create or update record if Unix working mode is selected for Oracle HTTP Server 11g) Absolute path to the top level directory where instances are configured. Maximum of 255 characters.</td>
</tr>
</tbody>
</table>
Scan Authentication

Oracle HTTP Server Record

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unix_inst_name={value}</td>
<td>(Optional) The Oracle HTTP server instance name. Maximum of 4000 characters.</td>
</tr>
</tbody>
</table>

**Windows Configuration**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>windows_home_path={value}</td>
<td>(Required to create or update record if Windows working mode is selected) The home directory path. Maximum of 255 characters.</td>
</tr>
<tr>
<td>windows_domain_path={value}</td>
<td>(Required to create or update record if Windows working mode is selected for Oracle HTTP Server 12c and higher) Absolute path to the top level directory where domains are configured. Maximum of 255 characters.</td>
</tr>
<tr>
<td>windows_inst_path={value}</td>
<td>(Required to create or update record if Windows working mode is selected for Oracle HTTP Server 11g) Absolute path to the top level directory where instances are configured. Maximum of 255 characters.</td>
</tr>
<tr>
<td>windows_inst_name={value}</td>
<td>(Optional) The Oracle HTTP server instance name. Maximum of 4000 characters.</td>
</tr>
</tbody>
</table>

**Sample - Create Oracle HTTP Server 11g Record(s) on Unix**

**API request:**

```bash
curl -u "USERNAME:PASSWORD" -S -H 'X-Requested-With:curl demo2' -d "action=create&title=Oracle_HTTP_Unix server_11&unix_home_path=/opt/Oracle/Middleware/Oracle_WT1&unix_inst_path=/opt/Oracle/Middleware/Oracle_WT1/instances/instance1&unix_inst_name=ohs1&ips=10.11.70.24" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/"
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-10-15T05:51:21Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>1530246</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Sample - Create Oracle HTTP Server 11g Record(s) on Windows

API request:

curl -u "USERNAME:PASSWORD" -S -H 'X-Requested-With:curl demo2' -d "action=create&title=Oracle_HTTP_Windows_server_11&windows_home_path=C:\Middleware\Oracle_WT1\instances\instance1&windows_inst_path=C:\Middleware\Oracle_WT1\instances\instance1&windows_inst_name=ohs1&ips=10.11.70.193" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-10-15T05:50:01Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>1530243</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

Sample - Create Oracle HTTP Server 12c Record(s) on Unix

API request:

curl -u "USERNAME:PASSWORD" -S -H 'X-Requested-With:curl demo2' -d "action=create&title=Oracle_HTTP_Unix_server_12&unix_home_path=/opt/Oracle/Middleware/Oracle_Home&unix_domain_path=/opt/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain&windows_inst_name=ohs1&ips=10.11.70.68" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-10-15T05:45:50Z</DATETIME>
    <BATCH_LIST>
      <BATCH>

454
Sample - Create Oracle HTTP Server 12c Record(s) on Windows

API request:

curl -u "USERNAME:PASSWORD" -S -H 'X-Requested-With:curl demo2' -d "action=create&title=Oracle_HTTP_Windows server_12&windows_home_path=C:\Oracle\Middleware\Oracle_Home&windows_domain_path=C:\Oracle\Middleware\Oracle_Home\user_projects\domains\base_domain&windows_inst_path=C:\Oracle\Middleware\Oracle_Home\instances\instance1&windows_inst_name=ohs1&ips=10.11.70.84" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-10-15T05:48:55Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>1530241</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

Sample - Update Oracle HTTP Server 11g Record(s) on Unix

API request:

curl -u "USERNAME:PASSWORD" -S -H 'X-Requested-With:curl demo2' -d "action=update&ids=1530246&unix_home_path=/opt/Oracle/Middleware/Oracle_WT1&unix_inst_path=/opt/Oracle/Middleware/Oracle_WT1/instances\instancename\unix_inst_name=ohs1&ips=10.11.70.24&comments=ohn unix"
auth record updated"
"https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-10-15T06:01:38Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>1530246</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Update Oracle HTTP Server 11g Record(s) on Windows

API request:

```bash
curl -u "USERNAME:PASSWORD" -S -H 'X-Requested-With:curl demo2' -d "action=update&ids=1530243&windows_home_path=C:\Middleware\Oracle_WT1&windows_inst_path=C:\Middleware\Oracle_WT1\instances\instance1 &windows_inst_name=ohs1&ips=10.11.70.193&comments=ohs wind auth record updated"
"https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-10-15T06:05:43Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>1530243</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Sample - Update Oracle HTTP Server 12c Record(s) on Unix

API request:
```
curl -u "USERNAME:PASSWORD" -S -H 'X-Requested-With:curl demo2' -d "action=update&updateids=1530234&unix_home_path=/opt/Oracle/Middleware/Oracle_Home&unix_domain_path=/opt/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain&windows_inst_name=ohs1&ips=10.11.70.68&comments=ohs unix auth record updated" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-10-15T06:14:31Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>1530234</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample - Update Oracle HTTP Server 12c Record(s) on Windows

API request:
```
curl -u "USERNAME:PASSWORD" -S -H 'X-Requested-With:curl demo2' -d "action=update&updateids=1530241&windows_home_path=C:\Oracle\Middleware\Oracle_Home\windows_domain_path=C:\Oracle\Middleware\Oracle_Home\user_projects\domains\base_domain\windows_inst_path=C:\Oracle\Middleware\Oracle_Home\instances\instance1\windows_inst_name=ohs1&ips=10.11.70.84&comments=ohs wind auth record updated" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
```
Sample - List Oracle HTTP Server Records with Basic Details

API request:


XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_ORACLE_HTTP_SERVER_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/auth_oracle_http_server_list_output.dtd">
<AUTH_ORACLE_HTTP_SERVER_LIST_OUTPUT>
  <AUTH_ORACLE_HTTP_SERVER_LIST>
    <AUTH_ORACLE_HTTP_SERVER>
      <ID>1505927</ID>
      <TITLE><![CDATA[Oracle_HTTP_Unix server]]></TITLE>
      <IP_SET>
        <IP>10.11.70.24</IP>
      </IP_SET>
      <UNIX>
        <HOME_PATH><![CDATA[/opt/Oracle/Middleware/Oracle_WT1]]></HOME_PATH>
        <DOMAIN_PATH><![CDATA[]]></DOMAIN_PATH>
        <INST_PATH><![CDATA[/opt/Oracle/Middleware/Oracle_WT1/instances/in...]}]}>
  </AUTH_ORACLE_HTTP_SERVER>
</AUTH_ORACLE_HTTP_SERVER_LIST>
</AUTH_ORACLE_HTTP_SERVER_LIST_OUTPUT>
Sample - List Oracle HTTP Server Records with All Details

API request:

```
curl -S -H 'X-Requested-With:curl demo2' -u "USERNAME:PASSWORD" -d "action=list&details=All&ids=1505927" "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_ORACLE_HTTP_SERVER_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/auth/oracle_http_server/auth_oracle_http_server_list_output.dtd">
<AUTH_ORACLE_HTTP_SERVER_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2019-10-04T07:29:33Z</DATETIME>
    <AUTH_ORACLE_HTTP_SERVER_LIST>
      <AUTH_ORACLE_HTTP_SERVER>
        <ID>1505927</ID>
        <TITLE><![CDATA[Oracle_HTTP_Unix server]]></TITLE>
        <IP_SET>
          <IP>10.11.70.24</IP>
        </IP_SET>
        <UNIX>
          <HOME_PATH><![CDATA[/opt/Oracle/Middleware/Oracle_WT1]]></HOME_PATH>
          <DOMAIN_PATH><![CDATA[]]></DOMAIN_PATH>
          <INST_PATH><![CDATA[/opt/Oracle/Middleware/Oracle_WT1/instances/instance1]]></INST_PATH>
        </UNIX>
      </AUTH_ORACLE_HTTP_SERVER>
    </AUTH_ORACLE_HTTP_SERVER_LIST>
  </RESPONSE>
</AUTH_ORACLE_HTTP_SERVER_LIST_OUTPUT>
```
Sample - Delete Oracle HTTP Server Record(s)

API request:
```
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2019-10-04T09:19:50Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID>1507609</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Scan Authentication
Oracle HTTP Server Record

</RESPONSE>
</BATCH_RETURN>

**DTDs for auth type “oracle_http_server”**

<platform API server>/api/2.0/fo/auth/auth_records.dtd

<platform API server>/api/2.0/fo/auth/oracle_http_server/auth_oracle_http_server_list_output.dtd
vCenter - ESXi Mapping Records

/\api/2.0/fo/auth/vcenter/vcenter_mapping/

[POST]

Input Parameters

The following table shows input parameters used for listing, importing and purging vCenter - ESXi mapping data.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>echo_request={0/1}</td>
<td>(Optional) Specifies whether to echo the request’s input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.</td>
</tr>
<tr>
<td>action={action}</td>
<td>(Required) One action (list, import or purge) required for the request.</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(Optional to list) Used to filter the XML output to show only vulnerabilities that have a QID number greater than or equal to a QID number you specify.</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(Optional to list) Used to filter the XML output to show only vulnerabilities that have a QID number less than or equal to a QID number you specify.</td>
</tr>
<tr>
<td>output_format={XML</td>
<td>CSV}</td>
</tr>
<tr>
<td>truncation_limit={value}</td>
<td>(Optional to list) Specifies the maximum number records listed per request.</td>
</tr>
<tr>
<td>vcenter_ip={value}</td>
<td>(Optional to list) Specifies the IP address of the vCenter.</td>
</tr>
<tr>
<td>esxi_ip={value}</td>
<td>(Optional to list) Specifies the IP address of the ESXi server.</td>
</tr>
<tr>
<td>network_id={1</td>
<td>0}</td>
</tr>
<tr>
<td>csv_data={value}</td>
<td>(Required to import and purge) The CSV data file containing the vCenter - ESXi mapping records that you want to add/purge. This parameter or xml_data must be specified. The parameters csv_data and xml_data cannot be specified in the same request.</td>
</tr>
<tr>
<td>xml_data={value}</td>
<td>(Required to import and purge) The XML data file containing the vCenter - ESXi mapping records that you want to add/purge. This parameter or csv_data must be specified. The parameters csv_data and xml_data cannot be specified in the same request.</td>
</tr>
</tbody>
</table>
Sample - List vCenter - ESXi Mapping in CSV Format

API request:
```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl'
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/?action=list"
```

OR
```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl'
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/?action=list&output_format=csv"
```

CSV output:
```
----BEGIN_RESPONSE_BODY_CSV
vCenter IP,ESXi IP,Mapping Data Source
"11.11.11.11","30.30.30.23","File"
"10.10.10.10","10.10.10.12","File"
----END_RESPONSE_BODY_CSV
----BEGIN_RESPONSE_FOOTER_CSV
"Status Message"
"Finished"
----END_RESPONSE_FOOTER_CSV
```

Sample - List vCenter - ESXi Mapping in XML Format

API request:
```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl'
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/?action=list&output_format=xml"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE VCENTER_ESXI_MAP_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/vce
nter_esxi_map_list_output.dtd">
<VCENTER_ESXI_MAP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-05-22T16:49:40Z</DATETIME>
    <VCENTER_ESXI_MAP_LIST>
      <VCENTER_ESXI_MAP>
        <VCENTER_IP>11.11.11.11</VCENTER_IP>
        <ESXI_IP>30.30.30.23</ESXI_IP>
        <MAPPING_DATA_SOURCE>File</MAPPING_DATA_SOURCE>
      </VCENTER_ESXI_MAP>
      <VCENTER_ESXI_MAP>
        <VCENTER_IP>10.10.10.10</VCENTER_IP>
        <ESXI_IP>10.10.10.12</ESXI_IP>
        <MAPPING_DATA_SOURCE>File</MAPPING_DATA_SOURCE>
      </VCENTER_ESXI_MAP>
    </VCENTER_ESXI_MAP_LIST>
  </RESPONSE>
</VCENTER_ESXI_MAP_LIST_OUTPUT>
```
DTD for vCenter - ESXi Mapping

<platform API server>/api/2.0/fo/auth/vcenter/vcenter_mapping/vcenter_esxi_map_list_output.dtd

Sample - Import vCenter - ESXi Mapping

You’ll be able to import vCenter - ESXi mapping in the CSV and XML format. You can provide CSV or XML data in API call or in the file.

CSV Data in API Call

Following is the sample API request when you want to import mapping using CSV data in API call.

API request:

curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' --data-binary "action=import&csv_data=vCenter IP,ESXi IP
10.10.10.10,10.10.10.11
10.10.10.10,10.10.10.12"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/

XML Data in API Call

Following is the sample API request when you want to import mapping using XML data in API call.

API request:

curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' --data-binary "action=import&xml_data=<VCENTER_ESXI_MAP_LIST><VCENTER_ESXI_MAP><VCENTER_IP>11.11.11.11</VCENTER_IP><ESXI_IP>22.22.22.22</ESXI_IP></VCENTER_ESXI_MAP><VCENTER_ESXI_MAP><VCENTER_IP>11.11.11.12</VCENTER_IP><ESXI_IP>22.22.22.23</ESXI_IP></VCENTER_ESXI_MAP></VCENTER_ESXI_MAP_LIST>"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/

CSV Data in File

Following is the sample API request when you want to import the mapping using a file containing CSV data. In the sample request, add.csv is a CSV data file.

API request:

curl -u "USERNAME:PASSWORD" -H 'X-Requested-with: curl' --data-binary "@add.csv"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/

Sample content of add.csv file:

action=import&csv_data=
vCenter IP,ESXi IP
10.10.10.10,20.20.20.20
10.10.10.10,20.20.20.21
10.10.10.10,20.20.20.22
11.11.11.11,30.30.30.23
12.12.12.12,40.40.40.24
XML Data in File

Following is the sample API request when you want to import the mapping using a file containing XML data. In the sample request, `add.xml` is a XML data file.

**API request:**

```bash
```

**Sample content of `add.xml` file:**

```xml
action=import&xml_data=
<?xml version="1.0" encoding="UTF-8" ?>
<VCENTER_ESXI_MAP_LIST>
  <VCENTER_ESXI_MAP>
    <VCENTER_IP>10.10.10.10</VCENTER_IP>
    <ESXI_IP>20.20.20.21</ESXI_IP>
  </VCENTER_ESXI_MAP>
  <VCENTER_ESXI_MAP>
    <VCENTER_IP>10.10.10.10</VCENTER_IP>
    <ESXI_IP>20.20.20.22</ESXI_IP>
  </VCENTER_ESXI_MAP>
</VCENTER_ESXI_MAP_LIST>
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2020-05-07T10:57:23Z</DATETIME>
    <TEXT>Successfully imported 2 records</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Purge vCenter - ESXi Mapping

You’ll be able to purge vCenter - ESXi mapping in the CSV and XML format. You can provide CSV or XML data in API call or in the file.

**CSV Data in API Call**

Following is the sample API request when you want to purge mapping using CSV data in API call.

**API request:**

```bash
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' --data-binary "action=purge&csv_data=vCenter IP,ESXi IP\n10.10.10.10,10.10.10.11%0A10.10.10.10,10.10.10.12"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/
```

465
XML Data in API Call
Following is the sample API request when you want to purge mapping using XML data in API call.

API request:
```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' --data-binary
"action=purge&xml_data=<VCENTER_ESXI_MAP_LIST><VCENTER_ESXI_MAP><VCENTER_IP>11.11.11.11</VCENTER_IP><ESXI_IP>22.22.22.22</ESXI_IP></VCENTER_ESXI_MAP><VCENTER_ESXI_MAP><VCENTER_IP>11.11.11.12</VCENTER_IP><ESXI_IP>22.22.22.23</ESXI_IP></VCENTER_ESXI_MAP></VCENTER_ESXI_MAP_LIST>"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/
```

CSV Data in File
Following is the sample API request when you want to purge the mapping using a file containing CSV data. In the sample request, `purge.csv` is a CSV data file.

API request:
```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-with: curl' --data-binary
"@purge.csv" 
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/
```

Sample content of `purge.csv` file:
```
action=purge&csv_data=
vCenter IP,ESXi IP
10.10.10.10,20.20.20.20
10.10.10.10,20.20.20.21
10.10.10.10,20.20.20.22
11.11.11.11,30.30.30.23
12.12.12.12,40.40.40.24
```

XML Data in File
Following is the sample API request when you want to purge the mapping using a file containing XML data. In the sample request, `purge.xml` is a XML data file.

API request:
```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-with: curl' --data-binary
"@purge.xml" 
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/
```

Sample content of `purge.xml` file:
```
<?xml version="1.0" encoding="UTF-8" ?>
<VCENTER_ESXI_MAP_LIST>
  <VCENTER_ESXI_MAP>
    <VCENTER_IP>10.10.10.10</VCENTER_IP>
    <ESXI_IP>20.20.20.21</ESXI_IP>
  </VCENTER_ESXI_MAP>
</VCENTER_ESXI_MAP_LIST>
```
<VCENTER_IP>10.10.10.10</VCENTER_IP>
<ESXI_IP>20.20.20.22</ESXI_IP>
</VCENTER_ESXI_MAP>
</VCENTER_ESXI_MAP_LIST>

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2020-05-07T10:57:23Z</DATETIME>
    <TEXT>Successfully purged 2 records</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```
# Vault Support

Set up and manage integration with third party password vaults, an option for authenticated scanning (e.g. trusted scanning).

## Vault summary

<table>
<thead>
<tr>
<th>Vault Support matrix</th>
<th>View supported vaults by OS and supported features (i.e. password, key passphrase, private key)</th>
</tr>
</thead>
</table>

## Vault settings

<table>
<thead>
<tr>
<th>Vault Definition</th>
<th>Use Authentication API (/api/2.0/fo/auth/*) to add vault definition in authentication records</th>
</tr>
</thead>
<tbody>
<tr>
<td>List Vaults</td>
<td>Use Vault API (/api/2.0/fo/vault) to list vault records</td>
</tr>
<tr>
<td>Manage Vaults</td>
<td>Use Vault API (/api/2.0/fo/vault) to create, edit, and delete vault records</td>
</tr>
</tbody>
</table>

## Vault Support matrix

Supported vaults by authentication type (OS/technology) and capability (password, private key, key passphrase, root delegation tool password). Use the vault name as shown when providing vault name using the Qualys API (i.e. vault_type=Quest Vault).

Vaults can be defined as part of authentication records using the Authentication API (/api/2.0/fo/auth/*) except as noted below. Some vaults can be defined using the Vault API (/api/2.0/fo/vault).

<table>
<thead>
<tr>
<th>password</th>
<th>private key</th>
<th>key passphrase</th>
<th>root delegation passwd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azure MS SQL (compliance scans only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARCON PAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Azure Key</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(UI support only)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BeyondTrust PBPS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CA Access Control</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk AIM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HashiCorp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lieberman ERPM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quest Vault</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thycotic Secret Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>private key</td>
<td>key passphrase</td>
<td>root delegation</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------</td>
<td>---------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><strong>Cisco</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARCON PAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Azure Key</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CA PAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk AIM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HashiCorp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thycotic Secret</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Checkpoint Firewall (compliance scans only)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARCON PAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Azure Key</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk AIM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thycotic Secret</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>IBM DB2</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARCON PAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CA Access Control</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk AIM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HashiCorp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lieberman ERPM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quest Vault</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thycotic Secret</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Infoblox</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Azure Key</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HashiCorp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BeyondTrust PBPS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thycotic Secret</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wallix AdminBastion (WAB)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MariaDB (compliance scans only)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARCON PAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Azure Key</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BeyondTrust PBPS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk AIM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HashiCorp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quest Vault</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thycotic Secret</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MongoDB</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>private key</td>
<td>key passphrase</td>
<td>root delegation</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ARCON PAM</td>
<td>Azure Key</td>
<td>Azure Key</td>
<td>CA Access Control</td>
</tr>
<tr>
<td>Azure Key</td>
<td>BeyondTrust PBPS</td>
<td>CyberArk AIM</td>
<td>CyberArk AIM</td>
</tr>
<tr>
<td>BeyondTrust PBPS</td>
<td>CyberArk AIM</td>
<td>HashiCorp</td>
<td>CyberArk PIM Suite</td>
</tr>
<tr>
<td>CA Access Control</td>
<td>CyberArk AIM</td>
<td>HashiCorp</td>
<td>HashiCorp</td>
</tr>
<tr>
<td>CyberArk AIM</td>
<td>CyberArk PIM Suite</td>
<td>Thycotic Secret Server</td>
<td>Hitachi ID PAM</td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
<td>Quest Vault</td>
<td>Quest Vault</td>
</tr>
<tr>
<td>HashiCorp</td>
<td>Lieberman ERPM</td>
<td>Quest Vault</td>
<td>Thycotic Secret Server</td>
</tr>
<tr>
<td>Thycotic Secret Server</td>
<td>Thycotic Secret Server</td>
<td>Thycotic Secret Server</td>
<td>Thycotic Secret Server</td>
</tr>
</tbody>
</table>

**MS SharePoint (compliance scans only)**

<table>
<thead>
<tr>
<th>ARCON PAM</th>
<th>Azure Key</th>
<th>BeyondTrust PBPS</th>
<th>CA Access Control</th>
<th>CA PAM</th>
<th>CyberArk AIM</th>
<th>CyberArk PIM Suite</th>
<th>HashiCorp</th>
<th>Lieberman ERPM</th>
<th>Quest Vault</th>
<th>Thycotic Secret Server</th>
<th>Thycotic Secret Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>(UI support only)</td>
<td>Azure Key</td>
<td>BeyondTrust PBPS</td>
<td>CA Access Control</td>
<td>CyberArk AIM</td>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
<td>Lieberman ERPM</td>
<td>Quest Vault</td>
<td>Thycotic Secret Server</td>
<td>Thycotic Secret Server</td>
<td></td>
</tr>
<tr>
<td>ARCON PAM</td>
<td>Azure Key</td>
<td>BeyondTrust PBPS</td>
<td>CA Access Control</td>
<td>CyberArk AIM</td>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
<td>Lieberman ERPM</td>
<td>Quest Vault</td>
<td>Thycotic Secret Server</td>
<td>Thycotic Secret Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>password</td>
<td>private key</td>
<td>key passphrase</td>
<td>root delegation passwd</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>-------------</td>
<td>----------------</td>
<td>------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Neo4j</strong></td>
<td>ARCON PAM Vault</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Azure Key</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BeyondTrust PBPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberArk AIM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HashiCorp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thycotic Secret</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Oracle</strong></td>
<td>ARCON PAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Azure Key</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BeyondTrust PBPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CA Access Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberArk AIM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HashiCorp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lieberman ERPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quest Vault</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thycotic Secret</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>(UI support only)</strong></td>
<td>BeyondTrust PBPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CA Access Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberArk AIM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lieberman ERPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quest Vault</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thycotic Secret</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Palo Alto Firewall</strong></td>
<td>Azure Key</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BeyondTrust PBPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberArk AIM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quest Vault</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thycotic Secret</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Pivotal Greenplum</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>private key</td>
<td>key passphrase</td>
<td>root delegation password</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------------</td>
<td>--------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARCON PAM</td>
<td>Azure Key</td>
<td>Azure Key</td>
<td>CA Access Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CA Access Control</td>
<td>BeyondTrust PBPS</td>
<td>CA Access Control</td>
<td>CA PAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk AIM</td>
<td>CA PAM</td>
<td>CyberArk AIM</td>
<td>CyberArk AIM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HashiCorp</td>
<td>Thycotic Secret Server</td>
<td>HashiCorp</td>
<td>Hitachi ID PAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hitachi ID PAM</td>
<td></td>
<td>Thycotic Secret Server</td>
<td>Lieberman ERPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quest Vault</td>
<td></td>
<td></td>
<td>Quest Vault</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thycotic Secret Server</td>
<td></td>
<td></td>
<td>Thycotic Secret Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PostgreSQL (compliance scans only)**

<table>
<thead>
<tr>
<th>password</th>
<th>private key</th>
<th>key passphrase</th>
<th>root delegation password</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCON PAM</td>
<td>Azure Key</td>
<td>Azure Key</td>
<td>CA Access Control</td>
</tr>
<tr>
<td>CA Access Control</td>
<td>BeyondTrust PBPS</td>
<td>CA Access Control</td>
<td>CA PAM</td>
</tr>
<tr>
<td>CyberArk AIM</td>
<td>CA PAM</td>
<td>CyberArk AIM</td>
<td>CyberArk AIM</td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
</tr>
<tr>
<td>HashiCorp</td>
<td>Thycotic Secret Server</td>
<td>HashiCorp</td>
<td>Hitachi ID PAM</td>
</tr>
<tr>
<td>Hitachi ID PAM</td>
<td></td>
<td>Thycotic Secret Server</td>
<td>Lieberman ERPM</td>
</tr>
<tr>
<td>Quest Vault</td>
<td></td>
<td></td>
<td>Quest Vault</td>
</tr>
<tr>
<td>Thycotic Secret Server</td>
<td></td>
<td></td>
<td>Thycotic Secret Server</td>
</tr>
</tbody>
</table>

**SAP Hana (compliance scans only)**

<table>
<thead>
<tr>
<th>password</th>
<th>private key</th>
<th>key passphrase</th>
<th>root delegation password</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCON PAM</td>
<td>Azure Key</td>
<td>Azure Key</td>
<td>CA Access Control</td>
</tr>
<tr>
<td>Azure Key</td>
<td>BeyondTrust PBPS</td>
<td>CA Access Control</td>
<td>CA PAM</td>
</tr>
<tr>
<td>CyberArk AIM</td>
<td>CA PAM</td>
<td>CyberArk AIM</td>
<td>CyberArk AIM</td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
</tr>
<tr>
<td>HashiCorp</td>
<td>Thycotic Secret Server</td>
<td>HashiCorp</td>
<td>Hitachi ID PAM</td>
</tr>
<tr>
<td>Hitachi ID PAM</td>
<td></td>
<td>Thycotic Secret Server</td>
<td>Lieberman ERPM</td>
</tr>
<tr>
<td>Quest Vault</td>
<td></td>
<td></td>
<td>Quest Vault</td>
</tr>
<tr>
<td>Thycotic Secret Server</td>
<td></td>
<td></td>
<td>Thycotic Secret Server</td>
</tr>
</tbody>
</table>

**SAP IQ (compliance scans only)**

<table>
<thead>
<tr>
<th>password</th>
<th>private key</th>
<th>key passphrase</th>
<th>root delegation password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arcon PAM</td>
<td>Azure Key</td>
<td>Azure Key</td>
<td>CA Access Control</td>
</tr>
<tr>
<td>Azure Key</td>
<td>BeyondTrust PBPS</td>
<td>CA Access Control</td>
<td>CA PAM</td>
</tr>
<tr>
<td>CyberArk AIM</td>
<td>CA PAM</td>
<td>CyberArk AIM</td>
<td>CyberArk AIM</td>
</tr>
<tr>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
<td>CyberArk PIM Suite</td>
<td>HashiCorp</td>
</tr>
<tr>
<td>HashiCorp</td>
<td>Thycotic Secret Server</td>
<td>HashiCorp</td>
<td>Hitachi ID PAM</td>
</tr>
<tr>
<td>Hitachi ID PAM</td>
<td></td>
<td>Thycotic Secret Server</td>
<td>Lieberman ERPM</td>
</tr>
<tr>
<td>Liberman ERPM</td>
<td></td>
<td></td>
<td>Quest Vault</td>
</tr>
<tr>
<td>Quest Vault</td>
<td></td>
<td></td>
<td>Thycotic Secret Server</td>
</tr>
<tr>
<td>Thycotic Secret Server</td>
<td></td>
<td></td>
<td>Wallix AdminBastion</td>
</tr>
<tr>
<td>password</td>
<td>private key</td>
<td>key passphrase</td>
<td>root delegation</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td><strong>Sybase (compliance scans only)</strong></td>
<td>ARCON PAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberArk AIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberArk PIM Suite</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HashiCorp</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lieberman ERPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quest Vault</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thycotic Secret</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Unix</strong></td>
<td>ARCON PAM</td>
<td></td>
<td>Azure Key</td>
</tr>
<tr>
<td></td>
<td>Azure Key</td>
<td></td>
<td>CA Access Control</td>
</tr>
<tr>
<td></td>
<td>BeyondTrust PBPS</td>
<td></td>
<td>CA PAM</td>
</tr>
<tr>
<td></td>
<td>CA PAM</td>
<td></td>
<td>CyberArk AIM</td>
</tr>
<tr>
<td></td>
<td>CyberArk AIM</td>
<td></td>
<td>CyberArk PIM Suite</td>
</tr>
<tr>
<td></td>
<td>CyberArk PIM Suite</td>
<td></td>
<td>HashiCorp</td>
</tr>
<tr>
<td></td>
<td>HashiCorp</td>
<td></td>
<td>Hitatchi ID PAM</td>
</tr>
<tr>
<td></td>
<td>Thycotic Secret Server</td>
<td></td>
<td>Lieberman ERPM</td>
</tr>
<tr>
<td></td>
<td>Server</td>
<td></td>
<td>Quest Vault</td>
</tr>
<tr>
<td></td>
<td>Wallix AdminBastion</td>
<td></td>
<td>Thycotic Secret Server</td>
</tr>
<tr>
<td><strong>VMware</strong></td>
<td>BeyondTrust PBPS</td>
<td></td>
<td>Azure Key</td>
</tr>
<tr>
<td></td>
<td>CA Access Control</td>
<td></td>
<td>CA Access Control</td>
</tr>
<tr>
<td></td>
<td>CyberArk AIM</td>
<td></td>
<td>CA PAM</td>
</tr>
<tr>
<td></td>
<td>CyberArk PIM Suite</td>
<td></td>
<td>CyberArk AIM</td>
</tr>
<tr>
<td></td>
<td>Lieberman ERPM</td>
<td></td>
<td>CyberArk PIM Suite</td>
</tr>
<tr>
<td></td>
<td>Quest Vault</td>
<td></td>
<td>HashiCorp</td>
</tr>
<tr>
<td></td>
<td>Thycotic Secret Server</td>
<td></td>
<td>Hitatchi ID PAM</td>
</tr>
<tr>
<td></td>
<td>Server</td>
<td></td>
<td>Lieberman ERPM</td>
</tr>
<tr>
<td></td>
<td>Wallix AdminBastion</td>
<td></td>
<td>Quest Vault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Thycotic Secret Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Wallix AdminBastion</td>
</tr>
</tbody>
</table>
Various record types support adding vault definition as part of authentication record settings. When supported these parameters are used to provide the vault definition in record settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>login_type={basic</td>
<td>vault} (Required only when you want to create or update vault information)</td>
</tr>
<tr>
<td>vault_id={value}</td>
<td>(Required only when action=create and login_type=vault)</td>
</tr>
<tr>
<td></td>
<td>For Windows, vault_id and password parameters are mutually exclusive and cannot be specified in the same request.</td>
</tr>
<tr>
<td></td>
<td>For Unix, vault_id and password, cleartext_password parameters are mutually exclusive and cannot be specified in the same request.</td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
vault_type={value} | (Required only when action=create and login_type=vault) Want to know what vaults support what technologies and capabilities? See Vault Support matrix. Choose one: ARCON PAM, Azure Key, BeyondTrust PBPS, CA Access Control, CA PAM, CyberArk AIM, CyberArk PIM Suite, HashiCorp, Hitachi ID PAM (no parameters specific to this vault type.), Lieberman ERPM, Quest Vault, Thycotic Secret Server, Wallix AdminBastion (WAB).

**ARCON PAM**

vault_service_type={value} | (Required if vault type is ARCON PAM) Specify a vault service type for authenticating to the vault and launching the scan on the host. This value is validated against the predefined list of service types.

**Azure Key**

ak_secret_name={value} | (Required if vault type is Azure Key) The secret name assigned to the secret stored in the vault.

**BeyondTrust PBPS**

system_name={value} | (Optional if vault type is BeyondTrust PBPS) The managed system name (also known as asset name). When not specified, we’ll attempt to auto-discover the system name at scan time.

account_name={value} | (Optional if vault type is BeyondTrust PBPS) The account name. When not specified, we’ll try the username specified in the authentication record.

**CA Access Control**

d_end_point_name={value} | (Required if vault type is CA Access Control) The End-Point name identifies a managed system, either a target for local accounts or a domain controller for domain accounts. An End-Point name is a user-defined value within your installation of CA Access Control Enterprise Management. The End-Point name entered in this record must match a pre-defined name exactly.

d_end_point_type={value} | (Required if vault type is CA Access Control) The End-Point type represents the method of access to the End-Point system. CA Access Control Enterprise Management uses pre-defined values for various methods and the End-Point type value must match a pre-defined value exactly. Examples: "Windows Agentless" (for Windows accounts) and "SSH Device" (for Unix via SSH).
Vault Support
Vault Definition

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>end_point_container={value}</td>
<td>(Required if vault type is CA Access Control) The End-Point container stores configuration values. CA Access Control Enterprise Management uses pre-defined values for various methods and the End-Point container value must match a pre-defined value exactly. Examples: &quot;Accounts&quot; (for Windows accounts) and &quot;SSH Accounts&quot; (for Unix via SSH).</td>
</tr>
</tbody>
</table>

CA PAM

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vault_app_name={value}</td>
<td>(Required) Application name as defined in the vault configuration for accessing a specific device.</td>
</tr>
<tr>
<td>vault_device_name={value}</td>
<td>(Optional) Specify the target device name defined in the vault configuration for which you want to retrieve the credentials. The device name can use one or more variables when defining the target name in order to match several targets that use the same naming convention.</td>
</tr>
</tbody>
</table>

You can use one or more variables when defining the device name in order to match several targets that use the same naming convention.

- ${ip} // The IP address of the target, i.e. 10.20.30.40.
- ${ip_dash} // The IP address of the target with dashes instead of dots, i.e. 10-20-30-40.
- ${dnshost} // The DNS host name of the target, i.e. host.domain.
- ${host} // The host name of the target, i.e. host before .domain.
- ${nbhost} // (Windows only) The NetBIOS host name of the target in upper-case, i.e. HOST_ABC.

Example, device-unix-${ip} will match these 3 devices: device-unix-10.50.60.70, device-unix-10.50.60.88 and device-unix-10.30.10.12.

Note: You must specify "vault_device_name" or "vault_device_host", but not both.
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vault_device_host={value}</td>
<td>(Optional) Specify the target device address defined in the vault configuration for which you want to retrieve the credentials. You can use one or more variables when defining the device host in order to match several targets that use the same naming convention.</td>
</tr>
</tbody>
</table>
|                               | ${ip} - The IP address of the target, i.e. 10.20.30.40.  
|                               | ${ip_dash} - The IP with dashes, i.e. 10-20-30-40.  
|                               | ${dnshost} - DNS hostname of the target, i.e. host.domain.  
|                               | ${host} - Hostname of the target, i.e. host before .domain.  
|                               | ${nbhost} - (Windows only) The NetBIOS name of the target in upper-case, i.e. HOST_ABC.  
|                               | For example, ${host}-${ip_dash} will match these 3 devices: host40-10-20-30-40, host80-10-50-60-70 and host12-10-30-10-12.  
|                               | Note: You must specify "vault_device_name" or "vault_device_host", but not both.                                                                                                                                                                                                            |
| folder={value}                | (Required if vault type is CyberArk AIM) Specify the name of the folder in the secure digital safe where the password to be used for authentication should be stored. The folder name can contain a maximum of 169 characters. Entering a trailing /, as in folder/, is optional (when specified, the service removes the trailing / and does not save it in the folder name). The maximum length of a folder name with a file name is 170 characters (the leading and/or trailing space in the input value will be removed). These special characters cannot be included in a folder name: / : * ? " < > | <tab> You can use one or more variables when defining the folder name in order to match several targets that use the same naming convention. |
|                               | ${ip} - The IP address of the target, i.e. 10.20.30.40.  
|                               | ${ip_dash} - The IP with dashes, i.e. 10-20-30-40.  
|                               | ${dnshost} - DNS hostname of the target, i.e. host.domain.  
|                               | ${host} - Hostname of the target, i.e. host before .domain.  
|                               | ${nbhost} - (Windows only) The NetBIOS name of the target in upper-case, i.e. HOST_ABC.  
<p>|                               | For example, ${host}-${ip_dash} will match these 3 targets: host40-10-20-30-40, host80-10-50-60-70 and host12-10-30-10-12.                                                                                                                                                                      |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file={value}</td>
<td>(Required if vault type is CyberArk AIM) Specify the name of the file in the secure digital safe where the password to be used for authentication should be stored. The file name can contain a maximum of 165 characters. The maximum length of a folder name plus a file name is 170 characters (the leading and/or trailing space in the input value will be removed). These special characters cannot be included in a file name: /*:?&quot;&lt;</td>
</tr>
<tr>
<td></td>
<td>You can use one or more variables when defining the file name in order to match several targets that use the same naming convention. ${ip} - The IP address of the target, i.e. 10.20.30.40. ${ip_dash} - The IP with dashes, i.e. 10-20-30-40. ${dnshost} - DNS hostname of the target, i.e. host.domain. ${host} - Hostname of the target, i.e. host before .domain. ${nbhost} - (Windows only) The NetBIOS name of the target in upper-case, i.e. HOST_ABC. For example, ${host}-${ip_dash} will match these 3 targets: host40-10-20-30-40, host80-10-50-60-70 and host12-10-30-10-12.</td>
</tr>
<tr>
<td>folder={value}</td>
<td>(Required if vault type is CyberArk PIM Suite) Specify the name of the folder in the secure digital safe where the password to be used for authentication should be stored. The folder name can contain a maximum of 169 characters. Entering a trailing /, as in folder/, is optional (when specified, the service removes the trailing / and does not save it in the folder name). The maximum length of a folder name with a file name is 170 characters (the leading and/or trailing space in the input value will be removed). These special characters cannot be included in a folder name: /*:?&quot;&lt;</td>
</tr>
<tr>
<td>file={value}</td>
<td>(Required if vault type is CyberArk PIM Suite) Specify the name of the file in the secure digital safe where the password to be used for authentication should be stored. The file name can contain a maximum of 165 characters. The maximum length of a folder name plus a file name is 170 characters (the leading and/or trailing space in the input value will be removed). These special characters cannot be included in a file name: /*:?&quot;&lt;</td>
</tr>
<tr>
<td>secret_kv_path={value}</td>
<td>(Optional if vault type is HashiCorp) The path of the secret engine. The default is &quot;secret/data&quot;. For a custom path, please provide path in the format &quot;path/to/secret/data&quot;. Note that we only support Key-Value Secret Engine version 2 to retrieve secrets from the HashiCorp Vault.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>secret_kv_name={value}</td>
<td>(Required if vault type is HashiCorp) The secret name which stores key-value pairs.</td>
</tr>
<tr>
<td>secret_kv_key={value}</td>
<td>(Required if vault type is HashiCorp) The key name for identifying a specific key-value pair.</td>
</tr>
</tbody>
</table>

**Lieberman ERPM**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_discover_system_name={0</td>
<td>1}</td>
</tr>
<tr>
<td>system_name_single_host={value}</td>
<td>(Required if vault type is Lieberman ERPM) Specify the system name that is needed to retrieve password for authentication. To specify system_name_single_host, ensure that auto discovery of system name is disabled (auto_discover_system_name=0). If auto discovery of system name is enabled (auto_discover_system_name=1), specifying system_name_single_host is invalid.</td>
</tr>
<tr>
<td>system_type={value}</td>
<td>(Required if vault type is Lieberman ERPM) A valid value is one of the following system type: auto, windows, unix, oracle, mssql, ldap, cisco, custom</td>
</tr>
<tr>
<td>custom_system_type={value}</td>
<td>(Required if vault type is Lieberman ERPM) Specify the custom system type name. custom_system_type is valid only when system_type=custom.</td>
</tr>
</tbody>
</table>

**Quest Vault**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>system_name={value}</td>
<td>(Required if vault type is Quest Vault) Specify the system name. During a scan we’ll perform a search for the system name and then retrieve the password. A single exact match of the system name must be found in order for authentication to be successful.</td>
</tr>
</tbody>
</table>

**Thycotic Secret Server**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret_name={value}</td>
<td>(Required if vault type is Thycotic Secret Server) Specify the secret name that contains the password to be used for authentication. The scanning engine will perform a search for the secret name and then get the password from the secret returned by the search. A single exact match of the secret name must be found in order for authentication to be successful. The secret name may contain a maximum of 256 characters, and must not contain multibyte characters.</td>
</tr>
<tr>
<td>authorization_name={value}</td>
<td>(Required if vault type is Wallix AdminBastion (WAB)) Specify the name of the authorization that enables secret retrieval from a group of targets.</td>
</tr>
<tr>
<td>target_name={value}</td>
<td>(Required if vault type is Wallix AdminBastion (WAB)) Specify the name of the target device using one of these formats:</td>
</tr>
<tr>
<td></td>
<td>user@global_WABdomain</td>
</tr>
<tr>
<td></td>
<td>user@local_WABdomain@device</td>
</tr>
<tr>
<td></td>
<td>where user is the user with access to the target, global_WABdomain is a domain name in a domain controller, local_WABdomain is a local domain, device is the device you want to scan</td>
</tr>
<tr>
<td></td>
<td>Use one or more variables in the target name to match several targets that use the same naming convention.</td>
</tr>
<tr>
<td></td>
<td>${ip} - The IP address of the target, i.e. 10.20.30.40.</td>
</tr>
<tr>
<td></td>
<td>${ip_dash} - The IP with dashes, i.e. 10-20-30-40.</td>
</tr>
<tr>
<td></td>
<td>${dnshost} - DNS hostname of the target, i.e. host.domain.</td>
</tr>
<tr>
<td></td>
<td>${host} - Hostname of the target, i.e. host before .domain.</td>
</tr>
<tr>
<td></td>
<td>${nbhost} - (Windows only) The NetBIOS name of the target in upper-case, i.e. HOST_ABC.</td>
</tr>
<tr>
<td></td>
<td>For example, the target name user@local_WABdomain@$ip will match these 3 devices:</td>
</tr>
<tr>
<td></td>
<td>10.50.60.70, 10.50.60.88 and 10.30.10.12.</td>
</tr>
</tbody>
</table>
List Vaults

The Authentication Vault API (resource /api/2.0/fo/vault/) allows you to list authentication vaults in your account. Use the parameter “action=list” to list the vaults

Permissions: Managers, Unit Managers and Scanners can view vaults and their settings.

API request:

curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -d "action=list" "https://qualysapi.qualys.com/api/2.0/fo/vault/"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_VAULT_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/vault/vault_output.dtd">
<AUTH_VAULT_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2014-09-12T13:55:57Z</DATETIME>
    <STATUS>Success</STATUS>
    <COUNT>13</COUNT>
    <AUTH_VAULTS>
      <AUTH_VAULT>
        <TITLE><![CDATA[added failover ip]]></TITLE>
        <VAULT_TYPE><![CDATA[CyberArk PIM Suite]]></VAULT_TYPE>
        <LAST_MODIFIED>
          <DATETIME>2014-02-13T12:05:21Z</DATETIME>
          <BY>quays_rn1</BY>
        </LAST_MODIFIED>
        <ID>1421</ID>
      </AUTH_VAULT>
      <AUTH_VAULT>
        <TITLE><![CDATA[added failover ip1]]></TITLE>
        <VAULT_TYPE><![CDATA[CyberArk PIM Suite]]></VAULT_TYPE>
        <LAST_MODIFIED>
          <DATETIME>2014-02-19T06:43:44Z</DATETIME>
          <BY>quays_rn1</BY>
        </LAST_MODIFIED>
        <ID></ID>
      </AUTH_VAULT>
    </AUTH_VAULTS>
  </RESPONSE>
</AUTH_VAULT_LIST_OUTPUT>
```
<ID>1441</ID>
</AUTH_VAULT>
<AUTH_VAULT>
<TITLE>
<![CDATA[Blue]]>
</TITLE>
<Vault_Type>
<![CDATA[CA Access Control]]>
</Vault_Type>
<Last_Modified>
<datetime>2013-09-21T05:26:32Z</datetime>
</Last_Modified>
</AUTH_VAULT>
</AUTH_VAULTS>
</RESPONSE>
</AUTH_VAULT_LIST_OUTPUT>

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Optional) Include vaults matching this title.</td>
</tr>
<tr>
<td>type={value}</td>
<td>(Optional) Include a certain vault type only. A valid value is: ARCON PAM Azure Key BeyondTrust PBPS CA Access Control CA PAM CyberArk AIM CyberArk PIM Suite HashiCorp Hitachi ID PAM Lieberman ERP M Quest Vault Thycotic Secret Server Wallix AdminBastion (WAB)</td>
</tr>
<tr>
<td>modified={date}</td>
<td>(Optional) Include vaults modified on or after a certain date/time, in this format: YYYY-MM-DD[THH:MM:SSZ] (UTC/GMT).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>orderby={value}</td>
<td>(Optional) Sort the vaults list by certain data. One of: “id”, “title”, “system_name”, “last_modified”, “last_modified_by”. A date must be specified in YYYYMM-DD[THH:MM:SSZ] format (UTC/GMT).</td>
</tr>
<tr>
<td>sortorder={asc</td>
<td>desc}</td>
</tr>
<tr>
<td>limit={value}</td>
<td>(Optional) The maximum number of vault records processed for the request, starting at the record number specified by the offset parameter. These parameters must be specified together: limit and offset. Limit value must always be greater than “0”. If you specify a value 0 for the parameter, the request will fail. When not specified, default limit is set to 1,000 vault records. You can specify a value less than or greater than the default.</td>
</tr>
<tr>
<td>offset={value}</td>
<td>(Optional) The starting vault record number, used only when the request includes the limit parameter.</td>
</tr>
</tbody>
</table>

More sample requests:

1) List all vaults, order vaults by system name
   
   ```
curl -H "X-Requested-With:API" -u "USERNAME:PASSWD" -d "action=list&orderby=system_name"
   "https://qualysapi.qualys.com/api/2.0/fo/vault/index.php/?"
   ```

2) List all vaults, order vaults by title in descending order
   
   ```
curl -H "X-Requested-With:API" -u "USERNAME:PASSWD" -d "action=list&sortorder=desc&title"
   "https://qualysapi.qualys.com/api/2.0/fo/vault/index.php/?"
   ```

3) List only 9th and 10th vault records
   
   ```
curl -H "X-Requested-With:API" -u "USERNAME:PASSWD" -d "action=list&limit=2&offset=9"
   "https://qualysapi.qualys.com/api/2.0/fo/vault/index.php/?"
   ```
Manage Vaults

The Authentication Vault API (resource `api/2.0/fo/vault`) allows you to manage authentication vaults (create, update, delete) as separate configurations.

Permissions: Managers can perform all functions (create, update, delete). Unit Managers can perform these functions if they are granted the permission “Create/edit authentication records/vaults”.

Create a new vault

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=create</td>
<td>(Required)</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required) The vault title.</td>
</tr>
<tr>
<td>type={value}</td>
<td>(Required) The vault type. A valid value is: ARCON PAM, Azure Key, BeyondTrust PBPS, CA Access Control, CA PAM, CyberArk AIM, CyberArk PIM Suite, HashiCorp, Hitachi ID PAM, Lieberman ERPM, Quest Vault, Thycotic Secret Server, Wallix AdminBastion (WAB)</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) User defined comments.</td>
</tr>
</tbody>
</table>

**API request:**

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=create&type=cyberark AIM&title=New-CyberArk-AIM&appid=CyberArk007&safe=Vaultsafe&url=https://afco.com&ssl_verify=1&cert=-----BEGIN+CERTIFICATE-----%0d%0aMIIEidCCAkcCAQEwDQYJKoZIwjdjELMAkGA1UEBhM%0d%0a-----END+CERTIFICATE-----&private_key_pwd=password&private_key=-----BEGIN+RSA+PRIVATE+KEY-----%0d%0aMIIEowIBAAKCAQEAmbSGAPwS662q5SsJ2XAA2mVvK0fXa%0d%0a-----END+RSA+PRIVATE+KEY-----" "https://qualysapi.qualys.com/api/2.0/fo/vault/index.php"
```
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2016-09-02T06:10:02Z</DATETIME>
        <TEXT>Success</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>7004</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

**Update vault settings**

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) A vault ID.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Optional) A new title to replace the existing title.</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Optional) User defined comments.</td>
</tr>
<tr>
<td>[vault settings]</td>
<td>“Tell me about vault settings”</td>
</tr>
</tbody>
</table>

**API request:**

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -X "POST" -d "id=14836922&server_address=10.10.10.10" "https://qualysapi.qualys.com/api/2.0/fo/vault/?action=update"
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2014-09-12T14:13:28Z</DATETIME>
        <TEXT>Success</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>7004</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```
View vault settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=view</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) A vault ID.</td>
</tr>
</tbody>
</table>

API request:


XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE VAULT_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/vault/vault_view.dtd">
<VAULT_OUTPUT>
  <RESPONSE>
    <DATETIME>2016-09-08T06:38:28Z</DATETIME>
    <VAULT_QUEST>
      <TITLE><![CDATA[New CyberArk AIM Vault]]></TITLE>
      <COMMENTS><![CDATA[]]></COMMENTS>
      <VAULT_TYPE><![CDATA[CyberArk AIM]]></VAULT_TYPE>
      <CREATED_ON>2016-09-07T07:09:34Z</CREATED_ON>
      <OWNER>user_john</OWNER>
      <LAST_MODIFIED>
        <DATETIME>2016-09-08T06:37:49Z</DATETIME>
        <BY>user_john</BY>
      </LAST_MODIFIED>
      <APPID><![CDATA[735435]]></APPID>
      <URL><![CDATA[https://afco.com]]></URL>
      <SSL_VERIFY><![CDATA[1]]></SSL_VERIFY>
      <SAFE><![CDATA[56908456904]]></SAFE>
      <ID>7004</ID>
    </VAULT_QUEST>
  </RESPONSE>
</VAULT_OUTPUT>
```
Delete a vault

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=view</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) A vault ID.</td>
</tr>
</tbody>
</table>

**API request:**
```bash
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -d "id=43463" "https://qualysapi.qualys.com/api/2.0/fo/vault/?action=delete"
```

**XML output:**
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-09-12T14:13:28Z</DATETIME>
    <TEXT>Success</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>Status</KEY>
        <VALUE>Deleted</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

**Tell me about vault settings**
The vault settings differ per vault type.

**ARCON PAM**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url={value}</td>
<td>(Required to create and optional to update vault) The HTTP or HTTPS URL to access the ARCON PAM Vault API. The HTTPS URL is required if the ssl_verify parameter is set 1.</td>
</tr>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required to create and optional to update vault) A username required to access the vault.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required to create and optional to update vault) A password required to access the vault.</td>
</tr>
</tbody>
</table>
### Azure Key

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url={value}</td>
<td>(Required to create and optional to update vault) The HTTP or HTTPS URL to access the Azure key Vault HTTP API. The HTTPS URL is required if the <code>ssl_verify</code> parameter is set 1.</td>
</tr>
<tr>
<td>app_id={value}</td>
<td>(Required to create and optional to update vault) The application ID associated with the application created in the Azure Key Vault.</td>
</tr>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
<tr>
<td>certificate={value}</td>
<td>(Required to create and optional to update vault) The client certificate for authentication. Enter the certificate block after the key block and be sure to include the first and last line (<code>-----BEGIN CERTIFICATE-----</code> and <code>-----END CERTIFICATE-----</code>). For a create/update request, if the <code>cert</code> parameter is specified, then the <code>private_key</code> parameter must also be specified.</td>
</tr>
<tr>
<td>private_key={value}</td>
<td>(Required to create and optional to update vault) The private key for authentication. Copy the contents of private key file (id_rsa) and be sure to include the first and last line (<code>-----BEGIN PRIVATE KEY-----</code> and <code>-----END PRIVATE KEY-----</code>).</td>
</tr>
<tr>
<td>passphrase={value}</td>
<td>(Optional) The private key passphrase is required if the private key is encrypted.</td>
</tr>
</tbody>
</table>

### BeyondTrust PBPS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>appkey={value}</td>
<td>(Required for new vault) The application key (alphanumeric string) for the BeyondTrust PBPS web services API. The maximum length is 128 bytes. A leading and/or trailing space or periods in the input value will be removed.</td>
</tr>
<tr>
<td>url={value}</td>
<td>(Required for new vault) The HTTP or HTTPS URL to access the BeyondTrust PBPS web services API.</td>
</tr>
<tr>
<td>ssl_verify={1</td>
<td>0}</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required for new vault) The user account that can call the BeyondTrust PBPS web services API. The maximum length is 64 characters. This special character cannot be included.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Optional) Specify a user password when required by the Application API Key configuration in BeyondTrust.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cert={value}</td>
<td>(Optional) Provide an X.509 client certificate with your private key when required by the Application API Key configuration in BeyondTrust. The certificate must be trusted by the PBPS web server. Enter the certificate block after the key block and be sure to include the first and last line (-----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----). For a create/update request, if the cert parameter is specified, then the private_key parameter must also be specified.</td>
</tr>
<tr>
<td>private_key={value}</td>
<td>(Optional) Specify the private key for authentication. Copy the contents of private key file (id_rsa) and be sure to include the first and last line (-----BEGIN PRIVATE KEY----- and -----END PRIVATE KEY-----). For a create/update request, if the private_key parameter is specified, then the cert parameter must also be specified.</td>
</tr>
<tr>
<td>private_key_pwd={value}</td>
<td>(Optional) Specify a password for your private key if it’s encrypted.</td>
</tr>
<tr>
<td>ca_url={value}</td>
<td>(Required for new vault) The HTTP or HTTPS URL of the CA Access Control web services, an API interface to your CA Access Control Enterprise Management installation. Note that the web services URL is different from the web management URL. Sample web services URL: <a href="http://caac126u-32-235.caac125.domain.com:18080/iam/TEWS6/ac">http://caac126u-32-235.caac125.domain.com:18080/iam/TEWS6/ac</a> Sample web management URL: <a href="http://caac126p-33-166.caac125.domain.com:18080/iam/ac/">http://caac126p-33-166.caac125.domain.com:18080/iam/ac/</a></td>
</tr>
<tr>
<td>ca_api_username={value}</td>
<td>(Required for new vault) The name of a user that is granted GetAccountPassword API permissions.</td>
</tr>
<tr>
<td>ca_ssl_verify={1</td>
<td>0}</td>
</tr>
<tr>
<td>ca_web_username={value}</td>
<td>(Optional) The web user name used to access Basic Authentication of the CA Access Control web server.</td>
</tr>
<tr>
<td>ca_web_password={value}</td>
<td>(Optional) The web password used to access Basic Authentication of the CA Access Control web server.</td>
</tr>
<tr>
<td>CA PAM</td>
<td></td>
</tr>
<tr>
<td>ssl_verify={0</td>
<td>1}</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>url={value}</td>
<td>(Required to create and optional to update vault) The HTTP or HTTPS URL to access the CA PAM Vault HTTP API.</td>
</tr>
<tr>
<td>apikey_name={value}</td>
<td>(Required to create and optional to update vault) The user account that can call the CA PAM Vault HTTP API.</td>
</tr>
<tr>
<td>apikey={value}</td>
<td>(Required to create and optional to update vault) The password for the user account that can call the CA PAM Vault HTTP API.</td>
</tr>
</tbody>
</table>

**CyberArk AIM**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>appid={value}</td>
<td>(Required) Application ID string defined by the customer. The application ID acts as an authenticator for our scanner to call CCP web services API. The maximum length of an application ID name is 128 bytes and the first 28 characters must be unique (leading and/or trailing space or periods in the input value will be removed). These restricted words cannot be included in an application ID: Users, Addresses, Areas, XUserRules, unknown, Locations, Safes, Schedule, VaultCategories, Builtin. These special characters cannot be included in an application ID: /: * ? &quot; &lt; &gt;</td>
</tr>
<tr>
<td>safe={value}</td>
<td>(Required) The name of the digital password safe. The safe name can contain a maximum of 28 characters (leading and/or trailing space in the input value will be removed). These special characters cannot be included in a safe name: /: * ? &quot; &lt; &gt;</td>
</tr>
<tr>
<td>url={value}</td>
<td>(Required) The HTTP or HTTPS URL over SSL protocols to access CyberArk's CCP web services.</td>
</tr>
<tr>
<td>ssl_verify={1</td>
<td>0}</td>
</tr>
<tr>
<td>cert={value}</td>
<td>(Optional) You must include an X.509 certificate with your private key. Enter the certificate block after the key block and be sure to include the first and last line (-----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----). For a create/update request, if the certificate parameter is specified, then the private_key parameter must also be specified.</td>
</tr>
<tr>
<td>private_key={value}</td>
<td>(Optional) Specify private key for authentication. Copy the contents of private key file (id_rsa) and be sure to include the first and last line (-----BEGIN PRIVATE KEY----- and ----END PRIVATE KEY-----). For a create/update request, if the private_key parameter is specified, then the certificate parameter must also be specified.</td>
</tr>
<tr>
<td>private_key_pwd={value}</td>
<td>(Optional) Specify a password for the encrypted private_key.</td>
</tr>
</tbody>
</table>
**CyberArk PIM Suite**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_address</td>
<td>(Required for new vault) The IP address of the vault server that stores system login credentials to be used.</td>
</tr>
<tr>
<td>port</td>
<td>(Optional) The port the vault server is running on. The port must be in the range 1025 to 65535. For a new vault the port is set to 1858 by default, if the port parameter is not specified.</td>
</tr>
<tr>
<td>safe</td>
<td>(Required for new vault) The name of the digital password safe. The safe name can contain a maximum of 28 characters (leading and/or trailing space in the input value will be removed). These special characters cannot be included in a safe name: \ / : * ? &quot; &lt; &gt; .</td>
</tr>
<tr>
<td>username</td>
<td>(Required for new vault) The username for an account with access to your CyberArk PIM Suite environment.</td>
</tr>
<tr>
<td>password</td>
<td>(Required for new vault) The password for an account with access to your CyberArk PIM Suite environment.</td>
</tr>
</tbody>
</table>

**HashiCorp**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>(Required) The HTTP or HTTPS URL to access the HashiCorp Vault HTTP API.</td>
</tr>
<tr>
<td>api_version</td>
<td>(Optional) The HashiCorp Vault HTTP API version. This is v1 by default, which is the only supported version.</td>
</tr>
<tr>
<td>ssl_verify</td>
<td>(Required to create and optional to update vault) When set to 1 (the default), our service will verify the SSL certificate of the web server to make sure the certificate is valid and trusted. When set to 0, our service will not verify the certificate of the web server.</td>
</tr>
<tr>
<td>auth_type</td>
<td>(Required to create vault, optional to update vault) HashiCorp Vault API supports three authentication types. First choose any one of the authentication method you want to use (Username/Password, Cert or App Role) and then provide login credentials for authenticating to the vault server via the HashiCorp Vault HTTP API.</td>
</tr>
<tr>
<td>auth_type=userpass</td>
<td>Choose this authentication method to authenticate to the vault server with a username and password combination. auth_type= {userpass} supports 3 parameters: path, username, password.</td>
</tr>
<tr>
<td>path</td>
<td>(Optional) The path for the Username/Password authentication method. The default path is auth/userpass but you can specify a custom path like auth/my-path.</td>
</tr>
<tr>
<td>username</td>
<td>(Required to create and update vault) The user account that can access the vault server.</td>
</tr>
<tr>
<td>password</td>
<td>(Required to create and update vault) The password for the user account.</td>
</tr>
</tbody>
</table>
### auth_type={cert}
Choose the this authentication method to authenticate to the vault server using SSL/TLS client certificates which are either signed by a CA (Certificate Authority) or self-signed. CA certificates are associated with a role name.

auth_type={cert} supports 5 parameters: path, role_name, cert, private_key, passphrase.

| path={value} | (Optional) The path for the Cert authentication method. The default path is auth/cert but you can specify a custom path like auth/my-path. |
| role_name={value} | (Required to create and update vault) The role associated with the CA certificate. |
| cert={value} | (Required to create and update vault) The client certificate for authentication. Enter the certificate block after the key block and be sure to include the first and last line (-----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----). For a create/update request, if the cert parameter is specified, then the private_key parameter must also be specified. |
| private_key={value} | (Required to create and update vault) The private key for authentication. Copy the contents of private key file (id_rsa) and be sure to include the first and last line (-----BEGIN PRIVATE KEY----- and -----END PRIVATE KEY-----). |
| passphrase{value} | (Optional) The private key passphrase, if the private key is encrypted. |

### auth_type={approle}
Choose the App Role authentication method to authenticate to the vault server with a vault-defined role.

auth_type={approle} supports 3 parameters: path, role_id, secret_id.

| path={value} | (Optional) The path for the App Role authentication method. The default path is auth/approle but you can specify a custom path like auth/my-path. |
| role_id={value} | (Required to create and update vault) The role ID of the App Role you want to use for authentication. |
| secret_id={value} | (Optional) The secret ID of the App Role you want to use for authentication. |

### Hitachi ID PAM
Hitachi ID PAM supports a parameter:

<p>| url={value} | (Required for new vault) The HTTP or HTTPS URL of the Hitachi ID PAM webservices. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username={value}</td>
<td>(Required for new vault) The username (ID) for the Hitachi ID PAM user account. To allow Qualys scanners to connect using this account, this user must have the following settings under Administrator information in the Hitachi ID Management Suite: 1) the privilege &quot;OTP IDAPI caller&quot; and 2) the value entered in the &quot;IP address with CIDR bitmask&quot; field must include the Qualys scanner IP addresses.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required for new vault) The password for the Hitachi ID PAM user account.</td>
</tr>
<tr>
<td>ssl_verify={1</td>
<td>0}</td>
</tr>
</tbody>
</table>

**Lieberman ERPM**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url={value}</td>
<td>(Required for new vault) The HTTP or HTTPS URL of the Lieberman ERPM server.</td>
</tr>
<tr>
<td>domain={value}</td>
<td>(Optional) A domain name if your Lieberman ERPM server is part of a domain.</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required for new vault) The username for the Lieberman ERPM server account.</td>
</tr>
<tr>
<td>password={value}</td>
<td>(Required) The password for the Lieberman ERPM server account.</td>
</tr>
<tr>
<td>ssl_verify={1</td>
<td>0}</td>
</tr>
</tbody>
</table>

**Quest Vault**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_address={value}</td>
<td>(Required for new vault) The IP address of the vault server, Quest One Privileged Password Manager.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Optional) The listing port of the vault server. For a new vault the port is set to 22 by default, if the port parameter is not specified.</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Required for new vault) The username to be used for SSH authentication. We recommend you create a dedicated user account for Qualys scanning. Using Quest/Dell 2.4 or higher, enter the key for the API user account you’ve created for use with our service. We support both API and CLI keys but recommend use of an API key.</td>
</tr>
<tr>
<td>access_key={value}</td>
<td>(Required for new vault) The DSA private key in PEM format for SSH authentication.</td>
</tr>
</tbody>
</table>

**Thycotic Secret Server**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url={value}</td>
<td>(Required for new vault) The HTTP or HTTPS URL of the Secret Server webservice. The URL may contain a maximum of 256 characters, and must not contain multibyte characters.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>username</td>
<td>(Required for new vault) The username for a Secret Server user. This user must have access to the secret names to be used for authentication.</td>
</tr>
<tr>
<td>password</td>
<td>(Required for new vault) The password for a Secret Server user.</td>
</tr>
<tr>
<td>domain</td>
<td>(Optional) Specify a fully qualified domain name if Secret Server is integrated with Active Directory. The domain may contain a maximum of 128 characters, and must not contain any multibyte characters.</td>
</tr>
<tr>
<td>url</td>
<td>(Required for new vault) The HTTP or HTTPS URL to access the WAB web services API.</td>
</tr>
<tr>
<td>ssl_verify</td>
<td>(Optional) When set to 1 (the default), our service will verify the SSL certificate of the web server to make sure the certificate is valid and trusted. When set to 0, our service will not verify the certificate of the web server.</td>
</tr>
<tr>
<td>username</td>
<td>(Required for new vault) The user account that can call the WAB web services API.</td>
</tr>
<tr>
<td>password</td>
<td>(Optional) The password for the user account that can call the WAB web services API. For a new vault, you must specify password or appkey. Both parameters cannot be specified in the same request.</td>
</tr>
<tr>
<td>appkey</td>
<td>(Optional) Your WAB REST API key (alpha-numeric value) for connecting to the WAB web services API.</td>
</tr>
</tbody>
</table>

**Wallix Admin Bastion (WAB)**

- Do not include leading or trailing periods or spaces.
- These characters are not allowed: `\ / : * ? " < > |`
- UTF-8 multibyte characters are not allowed.

For a new vault, you must specify password or appkey. Both parameters cannot be specified in the same request.
Assets

Manage the host assets you want to scan (internal and external facing) for vulnerabilities and compliance.

IP List | Add IPs | Update IPs
Host List | Host Update
Host List Detection | Normalized Data | Best Practices | Use Cases
Excluded Host List | Excluded Hosts Change History | Manage Excluded Hosts
Virtual Host List | Manage Virtual Hosts
Restricted IPs List | Manage Restricted IPs
Asset Group List | Manage Asset Groups
Purge Hosts
Patch List
**IP List**

/api/2.0/fo/asset/ip/?action=list

[GET]  [POST]

List IP addresses in the user account. By default, all hosts in the user account are included. Optional input parameters support filtering the list by IP addresses and host tracking method.

Permissions - Managers and Auditors view all assets in the subscription, Unit Managers view assets in their own business unit, Scanners and Readers view assets in their own account.

Express Lite - This API is available to Express Lite users.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required) A flag used to make an IP list request.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional) Show only certain IP addresses/ranges. One or more IPs/ranges may be specified. Multiple entries are comma separated. A host IP range is specified with a hyphen (for example, 10.10.10.44-10.10.10.90).</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional, and valid only when the Network Support feature is enabled for the user’s account) A non-Manager user can use this parameter to restrict the request to IP addresses in a certain custom network ID. For a Manager user, the output will be the same regardless of the network_id specified in the request because all IPs are part of all networks automatically and Managers have access to all IPs in all networks. Specify network_id along with tracking_method to filter the results.</td>
</tr>
<tr>
<td>tracking_method={value}</td>
<td>(Optional) Show only IP addresses/ranges which have a certain tracking method. Valid values: IP, DNS, NETBIOS.</td>
</tr>
</tbody>
</table>
Filter the output by module

Only interested in seeing IP addresses for VM, PC or CertView? Your request must include the `compliance_enabled` and `certview_enabled` parameters as described below.

To return only VM IP addresses, specify `compliance_enabled=0` and `certview_enabled=0`.

To return only PC IP addresses, specify `compliance_enabled=1` and `certview_enabled=0`.

To return only CertView IP addresses, specify `compliance_enabled=0` and `certview_enabled=1`.

To return both PC and CertView IP addresses, specify `compliance_enabled=1` and `certview_enabled=1`.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`compliance_enabled=[0</td>
<td>1]`</td>
</tr>
<tr>
<td>`certview_enabled=[0</td>
<td>1]`</td>
</tr>
</tbody>
</table>
Sample - List Host IPs

API request:
```
curl -H "X-Requested-With: Curl Sample" -b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure"
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/?action=list"
```

XML output:
```
<!DOCTYPE IP_LIST_OUTPUT SYSTEM
  "https://qualysapi.qualys.com/api/2.0/fo/asset/ip/
  ip_list_output.dtd">

<IP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-05-21T13:32:17Z</DATETIME>
    <IP_SET>
      <IP>123.123.45.0</IP>
      <IP_RANGE>123.124.45.0-123.124.45.255</IP_RANGE>
      <IP_RANGE>123.124.46.0-123.124.46.255</IP_RANGE>
      <IP_RANGE>123.124.47.0-123.124.47.255</IP_RANGE>
      <IP_RANGE>123.124.48.0-123.124.48.255</IP_RANGE>
    </IP_SET>
  </RESPONSE>
</IP_LIST_OUTPUT>
```

DTD
```
<platform API server>/api/2.0/fo/asset/ip/ip_list_output.dtd
```
Add IP addresses to the user’s subscription. Once added they are available for scanning and reporting.

Permissions - A Manager has permissions to add IP addresses. A Unit Manager can add IP addresses when the “Add assets” permission is enabled in their account. Users with other roles (Scanner, Reader, Auditor) do not have permissions to add IP addresses.

### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=add</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0</td>
<td>1]</td>
</tr>
<tr>
<td>ips=value -or- {POSTed CSV raw data}</td>
<td>(Required) The hosts you want to add to the subscription. IPs must be specified by using the “ips” parameter (using the POST method) or by uploading CSV raw data (using the POST method). To upload CSV raw data, specify --data-binary &lt;data&gt;. How to specify IP addresses. One or more IPs/ranges may be specified. Multiple IPs/ranges are comma separated. An IP range is specified with a hyphen (for example, 10.10.30.1-10.10.30.50). CIDR notation is supported.</td>
</tr>
<tr>
<td>tracking_method=value</td>
<td>(Optional) The tracking method is set to IP for IP address by default. To use another tracking method specify DNS or NETBIOS.</td>
</tr>
<tr>
<td>enable_vm=0</td>
<td>1] enable_pc=0</td>
</tr>
<tr>
<td>owner=value</td>
<td>(Optional) The owner of the host asset(s). The owner must be a Manager or a Unit Manager. A valid Unit Manager must have the “Add assets” permission and sufficient remaining IPs (maximum number of IPs that can be added to the Unit Manager’s business unit).</td>
</tr>
<tr>
<td>ud1=value ud2=value ud3=value</td>
<td>(Optional) Values for user-defined fields 1, 2 and 3. You can specify a maximum of 128 characters (ascii) for each field value.</td>
</tr>
<tr>
<td>comment=value</td>
<td>(Optional) User-defined comments.</td>
</tr>
</tbody>
</table>
Sample - Add IPs using POSTED data

API request:
```bash
curl -H "X-Requested-With: Curl" -H "Content-Type:text/csv"
-u "USERNAME:PASSWORD" --data-binary @ips_list.csv
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/?action=add&enable_vm=1&enable_pc=1&tracking_method=IP&owner=quay_es1"
```

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-08-07T01:21:03Z</DATETIME>
    <TEXT>IPs successfully added to Vulnerability Management/Compliance Management</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Add IPs using “ips” parameter

API request:
```bash
curl -H "X-Requested-With: demo" -u "USERNAME:PASSWORD" -X "POST"
-d "action=add&enable_vm=1&enable_pc=1&tracking_method=IP&owner=quay_es1"
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/"
```

DTD
```xml
<platform API server>/api/2.0/simple_return.dtd
```
Update IPs

/api/2.0/fo/asset/ip/?action=update

[POST]

Update IP addresses in the user's subscription.

**Good to Know**
- Host attributes you can update include tracking method (IP, DNS, NETBIOS), owner, user-defined fields (ud1, ud2, ud3), and comments.
- You cannot update an IP to use tracking method EC2 or AGENT. Also, if an IP is already tracked by EC2 or AGENT, you cannot change the tracking method to something else. We will skip the tracking method update in these cases.
- You can update multiple IPs/ranges in the same request. The host attribute changes will apply to all IPs included in the action.
- When the Network Support feature is enabled, you can update IPs in a custom network or in the Global Default Network. Only one network ID can be specified per update request. When a network ID is not specified in the request, we default to a value of 0 for Global Default Network.

**Permissions**
Managers have permission to update any IP, in any network. Sub-users (who have permission to update IPs) can update IPs for networks in their user scope. A Unit Manager can update IPs in asset groups assigned to their business unit. Users with other roles (Scanner, Reader, Auditor) do not have permission to update IP addresses.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=[0</td>
<td>1]</td>
</tr>
<tr>
<td>ips={value} -or-</td>
<td>(Required) The hosts within the subscription you want to update. IPs must be specified by using the &quot;ips&quot; parameter (using the POST method) or by uploading CSV raw data (using the POST method). To upload CSV raw data, specify -data-binary &lt;data&gt;. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.30.1-10.10.30.50). CIDR notation is supported.</td>
</tr>
<tr>
<td>{POSTed CSV raw data}</td>
<td></td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
`network_id={value}` | (Optional, and valid only when the Network Support feature is enabled for the user’s account) Restrict the request to a certain custom network by specifying the network ID. When unspecified, we default to "0" for Global Default Network.

`tracking_method={value}` | (Optional) To change to another tracking method specify IP for IP address, DNS or NETBIOS. You cannot change the tracking method to EC2 or AGENT. If an IP is already tracked by EC2 or AGENT, you cannot change the tracking method to something else.

`host_dns={value}` | (Optional) The DNS hostname for the IP you want to update. A single IP must be specified in the same request and the IP will only be updated if it matches the hostname specified.

`host_netbios={value}` | (Optional) The NetBIOS hostname for the IP you want to update. A single IP must be specified in the same request and the IP will only be updated if it matches the hostname specified.

`owner={value}` | (Optional) The owner of the host asset(s). The owner must be a Manager. Another user (Unit Manager, Scanner, Reader) can be the owner if the IP address is in the user’s account.

`ud1={value}`
`ud2={value}`
`ud3={value}` | (Optional) Values for user-defined fields 1, 2 and 3. You can specify a maximum of 128 characters (ascii) for each field value.

`comment={value}` | (Optional) User-defined comments.

### Sample - Add IPs and assign tracking method

**API request:**
```
curl -H "X-Requested-With: demo" -u "USERNAME:PASSWORD" -X "POST"  
-d "action=update&ips=10.10.200,10.10.23.40&tracking_method=DNS"  
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/"
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
 <RESPONSE>
  <DATETIME>2018-04-07T17:27:36Z</DATETIME>
  <TEXT>IPs successfully updated</TEXT>
 </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Update IP with matching NetBIOS name
IP 10.10.26.167 has multiple entries so we’re specifying the NetBIOS hostname in the request to identify which entry to update.

API request:
```
curl -H "X-Requested-With: demo" -u "USERNAME:PASSWORD" -X "POST"
-d "action=update&ips=10.10.26.167&host_netbios=ORA10105-WIN-25&comment=mycomment"
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/
```

Sample - Update IPs in custom network
(Applicable when the Network Support feature is enabled.) In this sample, network ID 2222 is specified in the request. The tracking method will be changed for the specified IPs in this network only.

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=update&network_id=2222&ips=10.10.10.200,10.10.23.40&tracking_method=DNS"
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/
```

Sample - Network ID is not in user's scope
(Applicable when the Network Support feature is enabled.) In this sample, the sub-user is trying to update an IP address in a network that is not in their scope.

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=update&network_id=55555&ips=10.10.10.10&comment=mycomment"
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2020-10-14T17:27:36Z</DATETIME>
    <CODE>1905</CODE>
    <TEXT>parameter network_id has invalid value: 55555 (No such network ID or not in user scope)</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Duplicate host error

For the request below we're updating IP 10.10.25.224. The duplicate host warning is returned because there are 2 asset records for IP 10.10.25.224.

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -X POST -d "action=update&ips=10.10.25.224&tracking_method=IP"
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE DUPLICATE_HOSTS_ERROR_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/asset/ip/duplicate_hosts_error.dtd">
<DUPLICATE_HOSTS_ERROR_OUTPUT>
  <RESPONSE>
    <CODE>1982</CODE>
    <DATETIME>2018-03-16T04:54:15Z</DATETIME>
    <WARNING>
      <TEXT>You cannot change the tracking method for the following host using the API since there are multiple scan data entries. This can happen when the host is resolved to different hostnames in different scan tasks. You'll need to change the tracking method using the UI. Use the URL to log into your account, edit the host and select another tracking method. At the prompt click Apply to save the most recent scan data and purge the other scan data.</TEXT>
      <DUPLICATE_HOSTS>
        <DUPLICATE_HOST>
          <IP>10.10.25.224</IP>
          <DNS_HOSTNAME>ora10105-win-25-224.qualys.com</DNS_HOSTNAME>
          <NETBIOS_HOSTNAME>ORA10105-WIN-25</NETBIOS_HOSTNAME>
          <LAST_SCANDATE>09/09/2016 at 13:35:29 (GMT)</LAST_SCANDATE>
          <TRACKING>DNS</TRACKING>
        </DUPLICATE_HOST>
      </DUPLICATE_HOSTS>
    </WARNING>
  </RESPONSE>
</DUPLICATE_HOSTS_ERROR_OUTPUT>
```

DTD for duplicate host error

```
<platform API server>/api/2.0/fo/asset/ip/duplicate_hosts_error.dtd"```
Host List

/api/2.0/fo/asset/host/?action=list

[GET]  [POST]

Download a list of scanned hosts in the user’s account. By default, all scanned hosts in the user account are included and basic information about each host is provided. Hosts in the XML output are sorted by host ID in ascending order.

The output of the Host List API is paginated. By default, a maximum of 1,000 host records are returned per request. You can customize the page size (i.e. the number of host records) by using the parameter “truncation_limit=10000” for instance. In this case the results will be return with pages of 10,000 host records.

The Host List API also shows the TruRisk score for each asset record in the API output and allows users to filter the output based on the TruRisk score.

The TruRisk score is the overall risk score assigned to the asset based on multiple contributing factors, including Asset Criticality Score (ACS), Risk (QID) scores for each severity level, and an auto assigned weighting factor (w) for each criticality level of QIDs.

TruRisk score has a range from 0 to 1000:

- Severe (850-1000)
- High (700-849)
- Medium (500-699)
- Low (0-499)

Permissions - Managers view all scanned hosts in subscription. Auditors view all scanned compliance hosts in subscription. Unit Managers view scanned hosts in user’s business unit. Scanners and Readers view scanned hosts in user’s account. Please note that this API only returns information for hosts that are assigned to each user through asset groups in VM/VMDR and PC.

For Unit Managers, Scanners, and Readers to view compliance hosts, the “Manage compliance” permission must be granted in the user’s account.

Express Lite - This API is available to Express Lite users.

### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required) A flag used to make a host list request.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>`show_asset_id=[0</td>
<td>1]`</td>
</tr>
<tr>
<td>`details=[Basic</td>
<td>Basic/AGs</td>
</tr>
<tr>
<td></td>
<td>Basic - (default) Show basic host information. Basic host information includes the host ID, IP address, tracking method, DNS and NetBIOS hostnames, and operating system.</td>
</tr>
<tr>
<td></td>
<td>Basic/AGs - Show basic host information plus asset group information. Asset group information includes the asset group ID and title.</td>
</tr>
<tr>
<td></td>
<td>All - Show all host information. All host information includes the basic host information plus the last vulnerability and compliance scan dates.</td>
</tr>
<tr>
<td></td>
<td>All/AGs - Show all host information plus asset group information. Asset group information includes the asset group ID and title.</td>
</tr>
<tr>
<td></td>
<td>None - Show only the host ID.</td>
</tr>
<tr>
<td><code>os_pattern={expression}</code></td>
<td>(Optional) Show only hosts which have an operating system matching a certain regular expression. An empty value cannot be specified. Use “%5E%24” to match empty string.</td>
</tr>
<tr>
<td></td>
<td>Important: The regular expression string you enter must follow the PCRE standard and it must be URL encoded.</td>
</tr>
<tr>
<td></td>
<td>Sample regular expression strings for matching OS names: Qualys API - Host List Detection API samples (GitHub, see sample 17)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>truncation_limit=</td>
<td>(Optional) Specify the maximum number of host records processed per request. When not specified, the truncation limit is set to 1000 host records. You may specify a value less than the default (1-999) or greater than the default (1001-1000000).</td>
</tr>
<tr>
<td></td>
<td>If the requested list identifies more host records than the truncation limit, then the XML output includes the &lt;WARNING&gt; element and the URL for making another request for the next batch of host records.</td>
</tr>
<tr>
<td></td>
<td>See example: [Qualys API - Host List API samples](GitHub, sample 3)</td>
</tr>
<tr>
<td></td>
<td>You can specify truncation_limit=0 for no truncation limit. This means that the output is not paginated and all the records are returned in a single output. WARNING: This can generate very large output and processing large XML files can consume a lot of resources on the client side. In this case it is recommended to use the pagination logic and parallel processing. The previous page can be processed while the next page is downloaded.</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional) Show only certain IP addresses/ranges. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.10.1-10.10.10.100).</td>
</tr>
<tr>
<td>ipv6={value}</td>
<td>(Optional) A valid IPv6 address. Multiple entries are comma separated. If ipv6 is used as filter parameter then other target input filter parameters are not accepted.</td>
</tr>
<tr>
<td>ag_ids={value}</td>
<td>(Optional) Show only hosts belonging to asset groups with certain IDs. One or more asset group IDs and/or ranges may be specified. Multiple entries are comma separated. A range is specified with a dash (for example, 386941-386945). Valid asset group IDs are required.</td>
</tr>
<tr>
<td>ag_titles={value}</td>
<td>(Optional) Show only hosts belonging to asset groups with certain strings in the asset group title. One or more asset group titles may be specified. Multiple entries are comma separated (for example, My+First+Asset+Group,Another+Asset+Group).</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Optional) Show only certain host IDs/ranges. One or more host IDs/ranges may be specified. Multiple entries are comma separated. A host ID range is specified with a hyphen (for example, 190-400). Valid host IDs are required.</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(Optional) Show only hosts which have a minimum host ID value. A valid host ID is required.</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(Optional) Show only hosts which have a maximum host ID value. A valid host ID is required.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>network_ids={value}</td>
<td>(Optional, and valid only when the Network Support feature is enabled for the user’s account) Restrict the request to certain custom network IDs. Multiple network IDs are comma separated.</td>
</tr>
<tr>
<td>compliance_enabled={0</td>
<td>1}</td>
</tr>
<tr>
<td>no_vm_scan_since={date}</td>
<td>(Optional) Show hosts not scanned since a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”. Permissions - An Auditor cannot specify this parameter.</td>
</tr>
<tr>
<td>no_compliance_scan_since={date}</td>
<td>(Optional) Show compliance hosts not scanned since a certain date and time (optional). This parameter is invalid for an Express Lite user. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”. Permissions - A sub-account (Unit Manager, Scanner or Reader) can specify this parameter only when the user is granted permissions to manage compliance information.</td>
</tr>
<tr>
<td>vm_scan_since={date}</td>
<td>(Optional) Show hosts that were last scanned for vulnerabilities since a certain date and time (optional). Hosts that were the target of a vulnerability scan since the date/time will be shown. Date/time is specified in this format: YYYY-MM-DD[THH:MM:SSZ] (UTC/GMT). Permissions: An Auditor cannot specify this parameter.</td>
</tr>
</tbody>
</table>

Date Filters
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>compliance_scan_since={date}</td>
<td>(Optional) Show hosts that were last scanned for compliance since a certain date and time (optional). Hosts that were the target of a compliance scan since the date/time will be shown. This parameter is invalid for an Express Lite user. Date/time is specified in this format: YYYY-MM-DD[THH:MM:SSZ] (UTC/GMT). Permissions: A sub-account (Unit Manager, Scanner or Reader) can specify this parameter only when the user is granted permissions to manage compliance information.</td>
</tr>
<tr>
<td>vm_processed_before={date}</td>
<td>(Optional) Show hosts with vulnerability scan results processed before a certain date and time. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td></td>
<td>Note: When a date is provided without a specific time, the query includes the data on or before the specified date, meaning that the specified date itself is included in the search query.</td>
</tr>
<tr>
<td>vm_processed_after={date}</td>
<td>(Optional) Show hosts with vulnerability scan results processed after a certain date and time. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td></td>
<td>Note: When a date is provided without a specific time, the query includes the data after the specified date, meaning that the specified date is excluded from the search query.</td>
</tr>
<tr>
<td>vm_scan_date_before={date}</td>
<td>(Optional) Show hosts with a vulnerability scan end date before a certain date and time. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td></td>
<td>Note: When a date is provided without a specific time, the query includes the data on or before the specified date, meaning that the specified date itself is included in the search query.</td>
</tr>
<tr>
<td>vm_scan_date_after={date}</td>
<td>(Optional) Show hosts with a vulnerability scan end date after a certain date and time. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td></td>
<td>Note: When a date is provided without a specific time, the query includes the data after the specified date, meaning that the specified date is excluded from the search query.</td>
</tr>
</tbody>
</table>
### Parameter | Description
---|---
vm_auth_scan_date_before={date} | (Optional) Show hosts with a successful authenticated vulnerability scan end date before a certain date and time. Specify the date in `YYYY-MM-DD[THH:MM:SSZ]` format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.

Note: When a date is provided without a specific time, the query includes the data on or before the specified date, meaning that the specified date itself is included in the search query.

vm_auth_scan_date_after={date} | (Optional) Show hosts with a successful authenticated vulnerability scan end date after a certain date and time. Specify the date in `YYYY-MM-DD[THH:MM:SSZ]` format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.

Note: When a date is provided without a specific time, the query includes the data after the specified date, meaning that the specified date is excluded from the search query.

scap_scan_since={date} | (Optional) Show hosts that were last scanned for SCAP since a certain date and time. Hosts that were the target of a SCAP scan since the date/time will be shown. This parameter is invalid for an Express Lite user. Valid date format is: `YYYY-MM-DD[THH:MM:SSZ]` format (UTC/GMT), like “2018-07-01” or “2018-01-25T23:12:00Z”.

no_scap_scan_since={date} | (Optional) Show hosts not scanned for SCAP since a certain date and time. This parameter is invalid for an Express Lite user. Valid date format is: `YYYY-MM-DD[THH:MM:SSZ]` format (UTC/GMT), like “2018-07-01” or “2018-01-25T23:12:00Z”.

### Asset Tags

use_tags={0|1} | (Optional) Specify 0 (the default) if you want to select hosts based on IP addresses/ranges and/or asset groups. Specify 1 if you want to select hosts based on asset tags.

tag_set_by={id|name} | (Optional when use_tags=1) Specify “id” (the default) to select a tag set by providing tag IDs. Specify “name” to select a tag set by providing tag names.

tag_include_selector={any|all} | (Optional when use_tags=1) Select “any” (the default) to include hosts that match at least one of the selected tags. Select “all” to include hosts that match all of the selected tags.

tag_exclude_selector={any|all} | (Optional when use_tags=1) Select “any” (the default) to exclude hosts that match at least one of the selected tags. Select “all” to exclude hosts that match all of the selected tags.

tag_set_include={value} | (Optional when use_tags=1) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag_set_exclude={value}</td>
<td>(Optional when use_tags=1) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>show_tags={0</td>
<td>1}</td>
</tr>
<tr>
<td><strong>Asset Risk Score (ARS)</strong></td>
<td><strong>Note:</strong> The ARS parameters will be retained for the next few releases. However, there will be a future update where these parameters will be removed with advance notification. You must start using the new TruRisk parameters mentioned below.</td>
</tr>
<tr>
<td>showArs={0</td>
<td>1}</td>
</tr>
<tr>
<td>ars_min={value}</td>
<td>(Optional) Show only asset records with an ARS value greater than or equal to the ARS min value specified. ars_min can only be specified when showArs=1. When ars_min and ars_max are specified in the same request, the ars_min value must be less than the ars_max value.</td>
</tr>
<tr>
<td>ars_max={value}</td>
<td>(Optional) Show only detection records with an ARS value less than or equal to the ARS max value specified. ars_max can only be specified when showArs=1. When ars_min and ars_max are specified in the same request, the ars_min value must be less than the ars_max value.</td>
</tr>
<tr>
<td>showArsFactors={0</td>
<td>1}</td>
</tr>
<tr>
<td><strong>TruRisk Score</strong></td>
<td></td>
</tr>
<tr>
<td>show_trurisk={0</td>
<td>1}</td>
</tr>
<tr>
<td>trurisk_min={value}</td>
<td>(Optional) Show only asset records with a TruRisk value greater than or equal to the TruRisk min value specified. trurisk_min can only be specified when show_trurisk=1. When trurisk_min and trurisk_max are specified in the same request, the trurisk_min value must be less than the trurisk_max value.</td>
</tr>
</tbody>
</table>
### Sample - List assets

**API request:**
```
curl --location --request GET
'<qualys_base_url>/api/2.0/fo/asset/host/?action=list&show_asset_id=1&host_metadata=all&details=All
--header 'X-Requested-With: curl'
--header 'Authorization: <token>'
```

**XML output:**
```
...
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM "<qualys_base_url>/api/2.0/fo/asset/host/dtd/list/output.dtd">
<HOST_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2023-04-25T03:52:05Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>32229</ID>
        <ASSET_ID>454545</ASSET_ID>
        <IP>10.14.28.129</IP>
        <TRACKING_METHOD>Cloud Agent</TRACKING_METHOD>
        <DNS>
          <![CDATA[pt-w1122h2u]]>
        </DNS>
        <DNS_DATA>
          <HOSTNAME>
            <![CDATA[pt-w1122h2u]]>
          </HOSTNAME>
          <DOMAIN />
          <FQDN />
        </DNS_DATA>
        <NETBIOS>
          <![CDATA[PT-W1122H2U]]>
        </NETBIOS>
        <OS>
          <![CDATA[Windows Microsoft Windows 11 Enterprise 10.0.22621 Build 22621]]>
        </OS>
        <QG_HOSTID>
          <![CDATA[c6656ff6-c4c3-40df-81b4-fffe361acf02]]>
        </QG_HOSTID>
        <FIRST_FOUND_DATE>2023-03-15T07:22:33Z</FIRST_FOUND_DATE>
        <LAST_BOOT>2023-03-15T07:22:33Z</LAST_BOOT>
        <SERIAL_NUMBER><![CDATA[hmhC53tK52oWfsv3]]></SERIAL_NUMBER>
        <HARDWARE_UUID><![CDATA[08b829fb-ff42-8e41-a2ae-1269ff6872b]]></HARDWARE_UUID>
        <LAST_ACTIVITY>2023-03-15T07:22:33Z</LAST_ACTIVITY>
        <AGENT_STATUS><![CDATA[Inventory Scan Complete]]></AGENT_STATUS>
      </HOST>
    </HOST_LIST>
  </RESPONSE>
</HOST_LIST_OUTPUT>
Sample - List assets based on scan end date, scan processed date

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/?action=list&
truncation_limit=10&details=All
/AGs&vm_scan_date_before=2017-09-
14T06:32:15Z&vm_auth_scan_date_before=2017-09-
14T06:32:15Z&vm_scan_date_after=2016-05-
12T06:32:15Z&vm_auth_scan_date_after=2016-05-
12T06:32:15Z&vm_processed_before=2017-09&scap_scan_since=2018-08-
29
-14T06:33:24Z&vm_processed_after=2016-09-12T06:33:24Z"-
12T06:33:24Z"
```

Sample - List assets with specific TruRisk score

**API request:**

```bash
curl --location
'qualys_base_url/api/2.0/fo/asset/host/?action=list&ips=10.115.126.190&
show_trurisk=1&trurisk_min=1&trurisk_max=1000&show_trurisk_factors=1'
--header 'X-Requested-With: curl demo2'
--header 'Authorization: Basic <token>'
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM
"qualys_base_url/api/2.0/fo/asset/host/dtd/list/output.dtd">
<HOST_LIST_OUTPUT>
 <RESPONSE>
  <DATETIME>2023-06-21T06:46:04Z</DATETIME>
  <HOST_LIST>
   <HOST>
    <ID>1800139</ID>
    <IP>10.115.126.190</IP>
    <ASSET_RISK_SCORE>680</ASSET_RISK_SCORE>
    <TRURISK_SCORE>680</TRURISK_SCORE>
    <ASSET_CRITICALITY_SCORE>4</ASSET_CRITICALITY_SCORE>
    <TRURISK_SCORE_FACTORS>
     <TRURISK_SCORE_FORMULA>4 *
     {(1.0*95*(14^0.01))+(0.6*74*(31^0.01))+(0.4*47*(73^0.01))+(0.2*33*(66^0.
     01))}</TRURISK_SCORE_FORMULA>
    <VULN_COUNT qds_severity="1">0</VULN_COUNT>
    <VULN_COUNT qds_severity="2">66</VULN_COUNT>
    <VULN_COUNT qds_severity="3">73</VULN_COUNT>
    <VULN_COUNT qds_severity="4">31</VULN_COUNT>
   </HOST>
  </HOST_LIST>
 </RESPONSE>
</HOST_LIST_OUTPUT>
```
Sample - List scanned assets with certain EC2 metadata

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"action=list&details=All&host_metadata=ec2&host_metadata_fields=region,accountId,instanceId"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/"
```

XML output:
```
<HOST_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/host_list_output.dtd">
<HOST_LIST_OUTPUT>
<RESPONSE>
<DATETIME>2017-04-15T09:50:46Z</DATETIME>
<HOST_LIST>
<HOST>
<ID>135151</ID>
<IP>10.97.5.247</IP>
</HOST>
```

<TRACKING_METHOD>EC2</TRACKING_METHOD>
<DNS><![CDATA[i-0bb87c3281243cdfd]]></DNS>
<EC2_INSTANCE_ID><![CDATA[i-0bb87c3281243cdfd]]></EC2_INSTANCE_ID>
<OS><![CDATA[Amazon Linux 2016.09]]></OS>
<METADATA>
<EC2>
  <ATTRIBUTE>
    <NAME><![CDATA[latest/dynamic/instance-identity/document/region]]></NAME>
    <LAST_STATUS>Success</LAST_STATUS>
    <VALUE><![CDATA[us-east-1]]></VALUE>
    <LAST_SUCCESS_DATE>2017-03-21T13:39:38Z</LAST_SUCCESS_DATE>
    <LAST_ERROR_DATE></LAST_ERROR_DATE>
    <LAST_ERROR><![CDATA[]]></LAST_ERROR>
  </ATTRIBUTE>
  <ATTRIBUTE>
    <NAME><![CDATA[latest/dynamic/instance-identity/document/accountId]]></NAME>
    <LAST_STATUS>Success</LAST_STATUS>
    <VALUE><![CDATA[205767712438]]></VALUE>
    <LAST_SUCCESS_DATE>2017-03-21T13:39:38Z</LAST_SUCCESS_DATE>
    <LAST_ERROR_DATE></LAST_ERROR_DATE>
    <LAST_ERROR><![CDATA[]]></LAST_ERROR>
  </ATTRIBUTE>
</EC2>
</METADATA>
<LAST_VULN_SCAN_DATETIME>2017-03-21T13:39:38Z</LAST_VULN_SCAN_DATETIME>
<LAST_VM_SCANNED_DATE>2017-03-21T13:39:38Z</LAST_VM_SCANNED_DATE>
<LAST_VM_SCANNED_DURATION>229</LAST_VM_SCANNED_DURATION>
<LAST_VM_AUTH_SCANNED_DATE>2017-03-21T13:39:38Z</LAST_VM_AUTH_SCANNED_DATE>
<LAST_VM_AUTH_SCANNED_DURATION>229</LAST_VM_AUTH_SCANNED_DURATION>
<LAST_COMPLIANCE_SCAN_DATETIME>2017-03-21T13:21:51Z</LAST_COMPLIANCE_SCAN_DATETIME>
</HOST>
</HOST_LIST>
</RESPONSE>
</HOST_LIST_OUTPUT>
Sample - Record Limit Exceeded Warning
In this case 1,000 host records are included in the XML output and the Warning message (shown below) indicates the URL you need to use to request the next 1,000 host records.

```xml
<RESPONSE>
  ...
  <WARNING>
    <CODE>1980</CODE>
    <TEXT>1000 record limit exceeded. Use URL to get next batch of results.</TEXT>
  </WARNING>
  <URL> <![CDATA[https://qualysapi.qualys.com/api/2.0/fo/asset/host/?action=list&id_min=2400356]]> </URL>
  </WARNING>
</RESPONSE>
```

**DTD**

```xml
<platform API server>/api/2.0/fo/asset/host/dtd/list/output.dtd
```
Host Update

`/api/2.0/fo/asset/host/?action=update`

[POST]

Here you can filter host assets based on input parameters and then you can update host attributes using new update parameters (new_tracking_method, new_owner, new_ud1, new_ud2, new_ud3, and new_comment).

**Good to Know**

- With host update API, you can update host attributes like tracking method (IP, DNS, NETBIOS), owner, user defined fields (ud1, ud2, ud3), and comments.
- You cannot update an IP to use tracking method EC2 or AGENT. Also, if an IP is already tracked by EC2 or AGENT, you cannot change the tracking method to something else. We will skip the tracking method update in these cases.

**Identify the hosts you want to update**

As part of the update request you’ll need to tell us which hosts you want to update. You can do this in a number of ways. You can simply specify the host IDs, or you can specify IP addresses, asset group IDs or asset group titles. When specifying IP addresses or asset groups, there are additional optional input parameters available.

Specify hosts using one of these combinations of input parameters:

- `ids` (required) only
- `ips` (required) with any of these optional parameters: `host_dns`, `host_netbios`, `network_id`, `network_name`, `tracking_method`
- `ag_ids` (required) with or without `tracking_method`
- `ag_titles` (required) with or without `tracking_method`

These input parameters are described in more detail below.

**Identify the changes you want to make**

Use new input parameters to tell us the host attributes you want to change. New input parameters include `new_tracking_method`, `new_owner`, `new_ud1`, `new_ud2`, `new_ud3`, and `new_comment`. The new values you specify will overwrite the existing values, and your changes will apply to all hosts included in the API request.

**Input Parameters**

Use these input parameters when updating hosts.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=[0</td>
<td>1]</td>
</tr>
</tbody>
</table>
### Host Filters

- **ids={value}**
  - Show only certain host IDs/ranges. One or more host IDs/ranges may be specified. Multiple entries are comma separated. A host ID range is specified with a hyphen (for example, 190-400). Valid host IDs are required.

- **ips={value} -or- {POSTed CSV raw data}**
  - The hosts within the subscription you want to update. IPs must be specified by using the “ips” parameter (using the POST method) or by uploading CSV raw data (using the POST method). To upload CSV raw data, specify --data-binary <data>.
  - One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.30.1-10.10.30.50). CIDR notation is supported.

- **network_id={value}**
  - (Valid only when the Network Support feature is enabled for the user’s account) Restrict the request to a certain custom network by specifying the network ID. When unspecified, we default to “0” for Global Default Network.

- **network_name={value}**
  - (Valid only when the Network Support feature is enabled for the user’s account) Restrict the request to a certain custom network by specifying the network name.

- **tracking_method={value}**
  - Show only IP addresses/ranges which have a certain tracking method.

- **host_dns={value}**
  - The DNS hostname for the IP you want to update. A single IP must be specified in the same request and the IP will only be updated if it matches the hostname specified.

- **host_netbios={value}**
  - The NetBIOS hostname for the IP you want to update. A single IP must be specified in the same request and the IP will only be updated if it matches the hostname specified.

### Host Changes

- **new_tracking_method={value}**
  - (Optional) Change the tracking method. Specify IP for IP address, DNS or NETBIOS. Note - You cannot change the tracking method to EC2 or AGENT. If an IP is already tracked by EC2 or AGENT, you cannot change the tracking method to something else.

- **new_owner={value}**
  - (Optional) Change the owner of the host asset(s). The owner must be a Manager. Another user (Unit Manager, Scanner, Reader) can be the owner if the IP address is in the user's account.

- **new_ud1={value}**
  - **new_ud2={value}**
  - **new_ud3={value}**
  - (Optional) Change values for user-defined fields 1, 2 and 3. You can specify a maximum of 128 characters (ascii) for each field value.

- **new_comment={value}**
  - (Optional) Change the user-defined comments. Specify new comments for the host asset(s).
Sample - Update Host Attributes with Host IDs

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo2"
"POST" -d "action=update&ids=2332017&new_tracking_method=DNS&new_ud1=Loc&new_ud2=Fun&new_ud3=AT&new_comment=API_Comment&new_owner=akreb_nb" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/asset/host/dtd/update/output.dtd">
<HOST_UPDATE_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-03-09T10:38:17Z</DATETIME>
    <TEXT>Assets successfully updated</TEXT>
  </RESPONSE>
</HOST_UPDATE_OUTPUT>

Sample - Update Host Attributes with IPs

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo2"
"POST" -d "action=update&ips=10.10.32.31&new_tracking_method=DNS&new_ud1=Loc&new_ud2=Fun&new_ud3=AT&new_comment=API_Comment&new_owner=akreb_nb" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/asset/host/dtd/update/output.dtd">
<HOST_UPDATE_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-03-09T06:03:42Z</DATETIME>
    <TEXT>Assets successfully updated</TEXT>
  </RESPONSE>
</HOST_UPDATE_OUTPUT>

Sample - Update Host Attributes with Asset Group IDs

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo2"
"POST" -d
"action=update&ag_ids=4580719&new_tracking_method=IP&new_ud1=Loc&
new_ud2=Fun&new_ud3=AT&new_comment=API_Comment&new_owner=akreb_nb"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/dtd/update/output.dtd">
<HOST_UPDATE_OUTPUT>
   <RESPONSE>
      <DATETIME>2021-03-09T10:39:11Z</DATETIME>
      <TEXT>Assets successfully updated</TEXT>
   </RESPONSE>
</HOST_UPDATE_OUTPUT>

Sample - Update Host Attributes with Asset Group Titles

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo2"
"POST" -d
"action=update&ag_titles=AG_Update&new_tracking_method=IP&new_ud1=
Loc&new_ud2=Fun&new_ud3=AT&new_comment=API_Comment&new_owner=akreb
_nb" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/dtd/update/output.dtd">
<HOST_UPDATE_OUTPUT>
   <RESPONSE>
      <DATETIME>2021-03-09T10:39:43Z</DATETIME>
      <TEXT>Assets successfully updated</TEXT>
   </RESPONSE>
</HOST_UPDATE_OUTPUT>

DTD for Host Update
<platform API server>/api/2.0/fo/asset/host/dtd/update/output.dtd
Host List Detection

/api/2.0/fo/asset/host/vm/detection/

[GET]  [POST]

Download a list of hosts with the hosts latest vulnerability data, based on the host based scan data available in the user’s account. This data brings a lot of value to customers because they provide the latest complete vulnerability status for the hosts (NEW, ACTIVE, FIXED, REOPENED) and history information.

The Host List VM Detection API also shows the Qualys Detection Score (QDS) for each detection record in the API output and allows users to filter the output based on the QDS.

The Qualys Detection Score (QDS) is assigned to vulnerabilities detected by Qualys. QDS is derived from multiple contributing factors, including vulnerability technical details (e.g. CVSS score), vulnerability temporal details (e.g. external threat intelligence like exploit code maturity), and remediation controls applied to mitigate the risk from the vulnerability.

QDS has a range from 1 to 100 with these severity levels:

- Critical (90-100)
- High (70-89)
- Medium (40-69)
- Low (1-39)

Permissions - Managers view all VM scanned hosts in subscription. Auditors have no permission to view VM scanned hosts. Unit Managers view VM scanned hosts in the user’s assigned business unit. Scanners and Readers view VM scanned hosts in the user’s account. Please note that this API only returns information for hosts that are assigned to each user through asset groups in VM/VMDR.

Express Lite - This API is available to Express Lite users.

Input Parameters

The input parameter action=list is required. All other input parameters are optional. Several filtering parameters are provided for filtering hosts and QIDs. When multiple filter parameters are specified, the service combines the effects of all the parameters in a way that corresponds to a logical "AND". So if two filter parameters are specified in the request, the service returns hosts that match both filters.

Quick Links: Detection Filters | Host Filters | QID Filters | Asset tags | Qualys Detection Score (QDS) | EC2/Azure/GCP metadata
## API Request

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0</td>
<td>1)</td>
</tr>
<tr>
<td>show_asset_id=0</td>
<td>1)</td>
</tr>
</tbody>
</table>
| include_vuln_type={confirmed|potential) | (Optional) Use to download vulnerability information based on their type, confirmed or potential. Specify:  
include_vuln_type=confirmed to download only confirmed vulnerabilities.  
include_vuln_type=potential to download only potential vulnerabilities. |

### Exceptional Scenarios

- After passing the parameter value include_vuln_type=confirmed, some potential vulnerabilities may show up in the API response. This happens when the vulnerability type for the QID is modified in the Qualys KnowledgeBase.

- After passing the parameter value include_vuln_type=potential, some confirmed vulnerabilities show up in the API response. This happens with the vulnerabilities assigned with a half red/half yellow severity level.

- This happens when the vulnerability type for the QID is changed while running the authenticated scan. It is because of various factors affecting scan results.
## Detection Filters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required) Show (echo) the request’s input parameters (names and values) in the output. When unspecified, parameters are not included in the output. Specify 1 to view parameters in the output.</td>
</tr>
<tr>
<td>echo_request=0</td>
<td>1</td>
</tr>
<tr>
<td>show_results=0</td>
<td>1</td>
</tr>
<tr>
<td>show_reopened_info=0</td>
<td>1</td>
</tr>
<tr>
<td>arf_kernel_filter=0</td>
<td>1</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| arf_service_filter=\{0|1|2|3|4\} | (Optional) Identify vulnerabilities found on running or non-running ports/services.  
When unspecified, vulnerabilities are not filtered based on running ports/services. `<AFFECT_RUNNING_SERVICE>` does not appear in the output.  
When set to 0, vulnerabilities are not filtered based on running ports/services. `<AFFECT_RUNNING_SERVICE>` appears in the output for service related vulnerabilities.  
When set to 1, exclude service related vulnerabilities that are exploitable (found on running ports/services). `<AFFECT_RUNNING_SERVICE>` appears in the output for service related vulnerabilities that have a value of 1.  
When set to 2, only include service related vulnerabilities that are exploitable (found on running ports/services). `<AFFECT_RUNNING_SERVICE>` appears in the output with a value of 0 for each detection.  
When set to 3, only include service related vulnerabilities that are not exploitable (found on non-running ports/services). `<AFFECT_RUNNING_SERVICE>` appears in the output with a value of 1 for each detection.  
When set to 4, only include service related vulnerabilities. `<AFFECT_RUNNING_SERVICE>` appears in the output with a value of 0 or 1 for each detection. |
| arf_config_filter=\{0|1|2|3|4\} | (Optional) Identify vulnerabilities that may or may not be exploitable due to the current host configuration.  
When unspecified, vulnerabilities are not filtered based on host configuration. `<AFFECT_EXPLOITABLE_CONFIG>` does not appear in the output.  
When set to 0, vulnerabilities are not filtered based on host configuration. `<AFFECT_EXPLOITABLE_CONFIG>` appears in the output for config related vulnerabilities.  
When set to 1, exclude vulnerabilities that are exploitable due to host configuration. `<AFFECT_EXPLOITABLE_CONFIG>` appears in the output for config related detections that have a value of 1.  
When set to 2, only include config related vulnerabilities that are exploitable. `<AFFECT_EXPLOITABLE_CONFIG>` appears in the output with a value of 0 for each detection.  
When set to 3, only include config related vulnerabilities that are not exploitable. `<AFFECT_EXPLOITABLE_CONFIG>` appears in the output with a value of 1 for each detection.  
When set to 4, only include config related vulnerabilities. `<AFFECT_EXPLOITABLE_CONFIG>` appears in the output with a value of 0 or 1 for each detection. |
### active_kernels_only

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Optional) Identify vulnerabilities related to running and non-running kernels in the output in the tag <code>&lt;AFFECT_RUNNING KERNEL&gt;</code>. Good to Know - It’s possible that multiple kernels are detected on a single Linux host. You’ll notice the scan results report the running kernel on each Linux host in Information Gathered QID 45097. When unspecified, vulnerabilities are not filtered based on kernel activity. <code>&lt;AFFECT_RUNNING KERNEL&gt;</code> does not appear in the output for kernel related vulnerabilities. When set to 0, vulnerabilities are not filtered based on kernel activity. <code>&lt;AFFECT_RUNNING KERNEL&gt;</code> appears in the output for kernel related vulnerabilities. When set to 1, exclude vulnerabilities found on non-running Linux kernels. <code>&lt;AFFECT_RUNNING KERNEL&gt;</code> appears in the output for kernel related vulnerabilities. When set to 2, only include vulnerabilities found on non-running Linux kernels. <code>&lt;AFFECT_RUNNING KERNEL&gt;</code> appears in the output with a value of 0 for all vulnerabilities. When set to 3, only include vulnerabilities found on running Linux kernels. <code>&lt;AFFECT_RUNNING KERNEL&gt;</code> appears in the output with a value of 1 for all vulnerabilities. Note that active_kernels_only is deprecated and will be removed in a future release. Please use arf_kernel_filter instead.</td>
</tr>
</tbody>
</table>
### Parameter: `output_format` values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>Specifies XML format for the output. When not specified, the output format is XML.</td>
</tr>
<tr>
<td>CSV</td>
<td>Specifies CSV format for the output. The output is structured in these sections: HEADER_CSV (lists input parameters specified during the list request if echo_request=1 is also specified), BODY_CSV (lists host records matching filters) and FOOTER_CSV (lists status messages and truncation details, if applicable).</td>
</tr>
<tr>
<td>CSV_NO_METADATA</td>
<td>Specifies CSV format for the output with no metadata. In this case, the output will not be structured with header, body and footer sections, and will not indicate whether the list is truncated.</td>
</tr>
<tr>
<td>CSV_NO_METADATA_MS_EXCEL</td>
<td>When specified we will use CSV format for the output with no metadata with MS Excel restrictions on the maximum length allowed for a string value in the output.</td>
</tr>
<tr>
<td>CSV_MS_EXCEL</td>
<td>When specified we will use CSV format for the output with MS Excel restriction on the maximum length allowed for a string value in the output. A value in the output will be truncated if the length of the value exceeds the maximum length supported in MS Excel.</td>
</tr>
</tbody>
</table>

### Parameter: `suppress_duplicated_data_from_csv` values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(Optional) By default or when set to 0, host details will be repeated in each line of detection information in the CSV output. When set to 1, host details will not be repeated (suppressed) in each detection line.</td>
</tr>
<tr>
<td>1</td>
<td>This parameter must be specified with: output_format=CSV or output_format=CSV_NO_METADATA.</td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
`truncation_limit={value}` | (Optional) Specifies the maximum number of host records processed per request. When not specified, the truncation limit is set to 1000 host records. You may specify a value less than the default (1-999) or greater than the default (1001-1000000). Specify 0 for no truncation limit. If the requested list identifies more host records than the truncation limit and `output_format=XML`, then the XML output includes the `<WARNING>` element and the URL for making another request for the next batch of host records. If the requested list identifies more host records than the truncation limit and `output_format=CSV`, then the CSV output includes “Truncated” in the FOOTER_CSV section and the URL for making another request for the next batch of host records. Check API samples (2, 4, 16) Qualys API - Host List Detection API samples (GitHub)

`max_days_since_detection_updated={value}` | (Optional) Show only detections whose detection status changed since some maximum number of days you specify. For detections that have never changed the maximum number of days is applied to the last detection date. One of these parameters may be specified in the same request: `detection_updated_since, max_days_since_detection_updated`

`detection_updated_since={value}` | (Optional) Show only detections whose detection status changed after a certain date and time. For detections that have never changed the date is applied to the last detection date. Valid date format is: `YYYY-MMDD[THH:MM:SSZ]` format (UTC/GMT), like “2017-02-15” or “2017-02-15T23:15:00Z”. Tip: You can use this parameter in conjunction with the `detection_updated_before` parameter to limit the detections shown to a specific date range. One of these parameters may be specified in the same request: `detection_updated_since, max_days_since_detection_updated`
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>detection_updated_before={value}</td>
<td>(Optional) Show only detections whose detection status changed before a certain date and time. Valid date format is: YYYY-MMDD[THH:MM:SSZ] format (UTC/GMT), like “2017-02-15” or “2017-02-15T23:15:00Z”. Tip: You can use this parameter in conjunction with the detection_updated_since parameter to limit the detections shown to a specific date range. One of these parameters may be specified in the same request: detection_updated_since, max_days_since_detection_updated</td>
</tr>
<tr>
<td>detection_processed_before={date}</td>
<td>(Optional) Show detections with vulnerability scan results processed before a certain date and time. Specify the date in YYYY-MMDD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td>detection_processed_after={date}</td>
<td>(Optional) Show detections with vulnerability scan results processed after a certain date and time. Specify the date in YYYY-MMDD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td>detection_last_tested_since={date}</td>
<td>(Optional) Show only detections that were last tested on or after a certain date and time. Valid date format is: YYYYMM-DD[THH:MM:SSZ] format (UTC/GMT), like “2018-07-01” or “2018-01-25T23:12:00Z”. You can use this parameter in conjunction with detection_last_tested_before or detection_last_tested_before_days to limit the detections shown to a date range. This parameter cannot be specified in the same request as detection_last_tested_since_days.</td>
</tr>
<tr>
<td>detection_last_tested_since_days={value}</td>
<td>(Optional) Show only detections that were last tested within the number of days you specify. For example, show detections last tested in the past 10 days. You can use this parameter in conjunction with detection_last_tested_before or detection_last_tested_before_days to limit the detections shown to a specific date range.</td>
</tr>
</tbody>
</table>
### Host List Detection

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>detection_last_tested_before</strong>={date}</td>
<td>(Optional) Show only detections that were last tested before a certain date and time. Valid date format is: YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2018-07-01” or “2018-01-25T23:12:00Z”. You can use this parameter in conjunction with detection_last_tested_since or detection_last_tested_since_days to limit the detections shown to a specific date range. This parameter cannot be specified in the same request as detection_last_tested_before_days.</td>
</tr>
<tr>
<td><strong>detection_last_tested_before_days</strong>={value}</td>
<td>(Optional) Show only detections that were last tested before the number of days you specify. For example, show detections last tested more than 30 days ago. You can use this parameter in conjunction with detection_last_tested_since or detection_last_tested_since_days to limit the detections shown to a specific date range. This parameter cannot be specified in the same request as detection_last_tested_before.</td>
</tr>
<tr>
<td><strong>include_ignored</strong>={0</td>
<td>1}</td>
</tr>
<tr>
<td><strong>include_disabled</strong>={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

### Host Filters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ids</strong>={value}</td>
<td>(Optional) Show only certain host IDs/ranges. One or more host IDs/ranges may be specified. Multiple entries are comma separated. A host ID range is specified with a hyphen (for example: 190-400). Valid host IDs are required.</td>
</tr>
<tr>
<td><strong>id_min</strong>={value}</td>
<td>(Optional) Show only hosts which have a minimum host ID value (for example: If id_min=44286508, then hosts with IDs 44286508 &amp; higher value are displayed).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(Optional) Show only hosts which have a maximum host ID value. A valid host ID is required for example: If id_max=44286508, then hosts with IDs 44286508 &amp; lower value are displayed.</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional) Show only certain IP addresses/ranges. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example: 10.10.10.1-10.10.10.100).</td>
</tr>
<tr>
<td>ipv6={value}</td>
<td>(Optional) A valid IPv6 address. Multiple entries are comma separated. If ipv6 is used as filter parameter then other target input filter parameters are not accepted.</td>
</tr>
<tr>
<td>ag_ids={value}</td>
<td>(Optional) Show only hosts belonging to asset groups with certain IDs. One or more asset group IDs and/or ranges may be specified. Multiple entries are comma separated. A range is specified with a dash (for example: 386941-386945). Valid asset group IDs are required.</td>
</tr>
<tr>
<td>ag_titles={value}</td>
<td>(Optional) Show only hosts belonging to asset groups with certain strings in the asset group title. One or more asset group titles may be specified. Multiple entries are comma separated (for example, My+First+Asset+Group,Another+Asset+Group).</td>
</tr>
<tr>
<td>network_ids={value}</td>
<td>(Optional, and valid only when the Network Support feature is enabled for the user’s account) Restrict the request to certain custom network IDs. Multiple network IDs are comma separated.</td>
</tr>
<tr>
<td>vm_scan_since={date}</td>
<td>(Optional) Show hosts scanned and processed since a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”.</td>
</tr>
<tr>
<td>no_vm_scan_since={date}</td>
<td>(Optional) Show hosts not scanned and processed since a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”.</td>
</tr>
<tr>
<td>max_days_since_last_vm_scan={value}</td>
<td>(Optional) Show only hosts scanned and processed in the past number of days, where the value is a number of days.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>vm_processed_before={date}</td>
<td>(Optional) Show hosts with vulnerability scan results processed before a certain date and time. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td>vm_processed_after={date}</td>
<td>(Optional) Show hosts with vulnerability scan results processed after a certain date and time. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td>vm_scan_date_before={date}</td>
<td>(Optional) Show hosts with a vulnerability scan end date before a certain date and time. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td>vm_scan_date_after={date}</td>
<td>(Optional) Show hosts with a vulnerability scan end date after a certain date and time. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td>vm_auth_scan_date_before={date}</td>
<td>(Optional) Show hosts with a successful authenticated vulnerability scan end date before a certain date and time. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td>vm_auth_scan_date_after={date}</td>
<td>(Optional) Show hosts with a successful authenticated vulnerability scan end date after a certain date and time. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2016-09-12” or “2016-09-12T23:15:00Z”.</td>
</tr>
<tr>
<td>status={value}</td>
<td>(Optional) Show only hosts with one or more of these status values: New, Active, Re-Opened, Fixed. Multiple status values are entered as a comma-separated list. If this parameter is not passed to the API, by default, the output contains detections with New, Active or Re-Opened &lt;STATUS&gt; only.</td>
</tr>
<tr>
<td>compliance_enabled={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>os_pattern={expression}</td>
<td>(Optional) Show only hosts which have an operating system matching a certain regular expression. An empty value cannot be specified. Use “%5E%24” to match empty string.</td>
</tr>
</tbody>
</table>

Important: The regular expression string you enter must follow the PCRE standard and it must be URL encoded.

Sample regular expression strings for matching OS names: [Qualys API - Host List Detection API samples (GitHub, see sample 17)](http://example.com)


For the PCRE syntax, see: [http://php.net/manual/en/reference.pcre.pattern.syntax.php](http://example.com)

### QID Filters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qids={value}</td>
<td>(Optional) Show only detection records with certain QIDs. One or more QIDs may be specified. A range is specified with a dash (for example: 68518-68522). Multiple entries are comma separated. Valid QIDs are required.</td>
</tr>
</tbody>
</table>

| severities={value}  | (Optional) Show only detection records which have certain severities. One or more levels may be specified. A range is specified with a dash (for example: 1-3). Multiple entries are comma separated.                       |

| filter_superseded_qids={0|1} | (Optional) When unspecified or set to 0, the XML output includes all QIDs even if they’ve been superseded. Specify 1 to filter out QIDs that have been superseded by another QID in the results. |

| show_igs={0|1}      | (Optional except as noted) Specify 1 to show detection records with information gathered along with confirmed vulnerabilities and potential vulnerabilities. Specify 0 (default) to hide information gathered. |

The show_igs parameter is required in one use case. The parameter show_igs=1 must be specified if both these conditions are met: 1) search lists are included using the parameter include_search_list_titles or include_search_list_ids, and 2) if the included search lists contain only information gathered.
### Host List Detection

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_search_list_titles=</td>
<td>(Optional) Show detection records only when a record’s QID is INCLUDED IN in one or more of the specified search list titles. One or more titles may be specified. Multiple titles are comma separated. This parameter cannot be specified with any of these parameters in the same request: qids, severities or include_search_list_ids.</td>
</tr>
<tr>
<td>exclude_search_list_titles=</td>
<td>(Optional) Show detection records only when a record’s QID is EXCLUDED from one or more of the specified search list titles. One or more titles may be specified. Multiple titles are comma separated. This parameter cannot be specified with any of these parameters in the same request: qids, severities or exclude_search_list_ids.</td>
</tr>
<tr>
<td>include_search_list_ids=</td>
<td>(Optional) Show detection records only when a record’s QID IS INCLUDED in one or more of the specified search list titles. One or more IDs may be specified. A range is specified with a dash (for example: 10-15). Multiple entries are comma separated. This parameter cannot be specified with any of these parameters in the same request: qids, severities or include_search_list_titles.</td>
</tr>
<tr>
<td>exclude_search_list_ids=</td>
<td>(Optional) Show detection records only when a record’s QID IS EXCLUDED from one or more of the specified search list titles. One or more IDs may be specified. A range is specified with a dash (for example: 40-42). Multiple entries are comma separated. This parameter cannot be specified with any of these parameters in the same request: qids, severities or exclude_search_list_titles.</td>
</tr>
</tbody>
</table>

### Asset tags

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>use_tags={0</td>
<td>1}</td>
</tr>
<tr>
<td>tag_set_by={id</td>
<td>name}</td>
</tr>
<tr>
<td>tag_include_selector={any</td>
<td>all}</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tag_exclude_selector=\{any\}[all]</code></td>
<td>(Optional when <code>use_tags=1</code>) Select “any” (the default) to exclude hosts that match at least one of the selected tags. Select “all” to exclude hosts that match all of the selected tags.</td>
</tr>
<tr>
<td><code>tag_set_include={value}</code></td>
<td>(Optional when <code>use_tags=1</code>) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.</td>
</tr>
<tr>
<td><code>tag_set_exclude={value}</code></td>
<td>(Optional when <code>use_tags=1</code>) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>`show_tags=[0</td>
<td>1]`</td>
</tr>
</tbody>
</table>
### Qualys Detection Score (QDS)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show_qds={0</td>
<td>1}</td>
</tr>
<tr>
<td>qds_min={value}</td>
<td>(Optional) Show only detection records with a QDS value greater than or equal to the QDS min value specified. qds_min can only be specified when show_qds=1. When qds_min and qds_max are specified in the same request, the qds_min value must be less than the qds_max value.</td>
</tr>
<tr>
<td>qds_max={value}</td>
<td>(Optional) Show only detection records with a QDS value less than or equal to the QDS max value specified. qds_max can only be specified when show_qds=1. When qds_min and qds_max are specified in the same request, the qds_min value must be less than the qds_max value.</td>
</tr>
<tr>
<td>show_qds_factors={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

### EC2/Azure/GCP metadata

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_metadata={value}</td>
<td>(Optional) Specify &quot;all&quot; to list all cloud assets with their metadata or specify the name of the cloud provider to show only the assets managed by the cloud provider. Valid values: all, ec2, google, azure</td>
</tr>
<tr>
<td>host_metadata_fields={value1,value2}</td>
<td>(Optional when host_metadata is specified) Specify metadata fields to only return data for certain attributes.</td>
</tr>
<tr>
<td></td>
<td>To retrieve information about the host instance type, use the metadata field, &quot;instance-type&quot;.</td>
</tr>
<tr>
<td>show_cloud_tags={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
Keep Alive Mechanism

The service uses a “keep alive” mechanism to maintain an open connection to the Qualys server for the duration of the host detection list API request. To keep the connection alive, the service sends some “dummy” data back to the client every 30 to 40 seconds if no “real” data has been sent already by the API during that time.

In XML output, this “dummy” data appears as a “<!-- keep-alive -->” line (since comments should be safely ignored by downstream XML parsers).

In CSV and CSV_NO_METADATA output, this “dummy” data appears as a <CR><LF> (carriage return, linefeed) pair (since empty lines clearly do not contain any CSV data).

Sample - Download a list of hosts with the latest vulnerability data, based on vulnerability type, confirmed

**API request:**
```
curl --location
  '<qualys_base_url>/api/2.0/fo/asset/host/vm/detection/?action=list
  &ips=11.111.11.111&include_vuln_type=confirmed' 
  --header 'X-Requested-With: curl'
  --header 'Authorization: Basic <encoded_username:password string>'
```

**XML output:**
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_VM_DETECTION_OUTPUT SYSTEM 
"<qualys_base_url>/api/2.0/fo/asset/host/vm/detection/dtd/output.dtd">
<HOST_LIST_VM_DETECTION_OUTPUT>
  <RESPONSE>
    <DATETIME>2023-08-16T13:34:18Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>524323</ID>
        <cloud_tag_fields>({value1, value2})
        
        (Optional when show_cloud_tags is specified) Specify cloud tags or cloud tag and name combinations to only return information for specified cloud tags. A cloud tag name and value combination is specified with a colon (for example:SomeTag:AY_ec2). For each cloud tag, we show the cloud tag’s name, its value, and last success date (the tag last success date/time, fetched from instance).

        If this parameter is not specified and "show_cloud_tags" is set to 1, we will show all the cloud provider tags for the assets.
      </HOST>
    </HOST_LIST>
  </RESPONSE>
</HOST_LIST_VM_DETECTION_OUTPUT>
```
Sample - Download a list of hosts with the latest vulnerability data, based on vulnerability type, potential

API request:

curl --location
'qualys_base_url/api/2.0/fo/asset/host/vm/detection/?action=list&ips=11.111.111.111&include_vuln_type=potential' \
--header 'X-Requested-With: curl'
--header 'Authorization: Basic <encoded username:password string>,'

XML output:
<xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_VM_DETECTION_OUTPUT SYSTEM "<qualys_base_url>/api/2.0/fo/asset/host/vm/detection/dtd/output.dtd">
<HOST_LIST_VM_DETECTION_OUTPUT>
  <RESPONSE>
    <DATETIME>2023-08-16T11:51:13Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>524323</ID>
        <IP>11.110.11.111</IP>
        <TRACKING_METHOD>IP</TRACKING_METHOD>
        <OS>
          <![CDATA[Linux 2.4-2.6 / Embedded Device / F5 Networks Big-IP]]>
        </OS>
        <LAST_SCAN_DATETIME>2019-11-08T09:15:14Z</LAST_SCAN_DATETIME>
        <LAST_VM_SCANNED_DATE>2019-11-08T06:37:29Z</LAST_VM_SCANNED_DATE>
        <DETECTION_LIST>
          <DETECTION>
            <UNIQUE_VULN_ID>3978356</UNIQUE_VULN_ID>
            <QID>38469</QID>
            <TYPE>Potential</TYPE>
            <SEVERITY>2</SEVERITY>
            <SSL>0</SSL>
            <RESULTS>
              <![CDATA[SSH-2.0-OpenSSH_3.9p1-AuthSelect-SecurID-log]]>
            </RESULTS>
            <STATUS>New</STATUS>
            <FIRST_FOUND_DATETIME>2019-11-08T06:37:29Z</FIRST_FOUND_DATETIME>
            <LAST_FOUND_DATETIME>2019-11-08T06:37:29Z</LAST_FOUND_DATETIME>
            <TIMES_FOUND>1</TIMES_FOUND>
            <LAST_TEST_DATETIME>2019-11-08T06:37:29Z</LAST_TEST_DATETIME>
            <LAST_UPDATE_DATETIME>2019-11-08T09:15:14Z</LAST_UPDATE_DATETIME>
            <IS_IGNORED>0</IS_IGNORED>
            <IS_DISABLED>0</IS_DISABLED>
            <LAST_PROCESSED_DATETIME>2019-11-08T09:15:14Z</LAST_PROCESSED_DATETIME>
          </DETECTION>
        </DETECTION_LIST>
      </HOST>
    </HOST_LIST>
  </RESPONSE>
</HOST_LIST_VM_DETECTION_OUTPUT>
Sample - List VM scanned hosts

API request:
```
curl -u "username:password" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?
action=list"
```

XML output:
```
<HOST_LIST_VM_DETECTION_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-04-26T11:25:58Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>6506432</ID>
        <IP>10.10.10.11</IP>
        <TRACKING_METHOD>IP</TRACKING_METHOD>
        <OS><![CDATA[Windows 2008 R2 Enterprise Service Pack 1]]></OS>
        <DNS><![CDATA[2k8r2-u-10-11.sample.qualys.com]]></DNS>
        <DNS_DATA>
          <HOSTNAME><![CDATA[2k8r2-u-10-11]]></HOSTNAME>
          <DOMAIN><![CDATA[sample.qualys.com]]></DOMAIN>
          <FQDN><![CDATA[2k8r2-u-10-11.sample.qualys.com]]></FQDN>
        </DNS_DATA>
        <NETBIOS><![CDATA[2K8R2-U-10-11]]></NETBIOS>
        <LAST_SCAN_DATETIME>2018-04-13T03:49:05Z</LAST_SCAN_DATETIME>
        <LAST_VM_SCANNED_DATE>2018-04-13T03:48:50Z</LAST_VM_SCANNED_DATE>
        <LAST_VM_SCANNED_DURATION>352</LAST_VM_SCANNED_DURATION>
        <DETECTION_LIST>
          <DETECTION>
            <UNIQUE_VULN_ID>52664</UNIQUE_VULN_ID>
          </DETECTION>
        </DETECTION_LIST>
      </HOST>
    </HOST_LIST>
  </RESPONSE>
</HOST_LIST_VM_DETECTION_OUTPUT>
```
<QID>38170</QID>
<Type>Confirmed</Type>
<Severity>2</Severity>
<Port>3389</Port>
<Protocol>tcp</Protocol>
<SSL>1</SSL>
<RESULTS><![CDATA[Certificate #0 CN=2k8r2-u-10-11
(2k8r2-u-10-11) doesn't resolve]]></RESULTS>
<Status>Active</Status>
<First_Found_Datetime>2018-01-26T04:45:50Z</First_Found_Datetime>
<Last_Found_Datetime>2018-04-13T03:48:50Z</Last_Found_Datetime>
<Times_Found>111</Times_Found>
<Last_Test_Datetime>2018-04-13T03:48:50Z</Last_Test_Datetime>
<Last_Update_Datetime>2018-04-13T03:49:05Z</Last_Update_Datetime>
<Is_Ignored>0</Is_Ignored>
<Is_Disabled>0</Is_Disabled>
<Last_Processed_Datetime>2018-04-13T03:49:05Z</Last_Processed_Datetime>
</Detection>

<Detection>
<Unique_Vuln_Id>51634</Unique_Vuln_Id>
<QID>38173</QID>
<Type>Confirmed</Type>
<Severity>2</Severity>
<Port>3389</Port>
<Protocol>tcp</Protocol>
<SSL>1</SSL>
<RESULTS><![CDATA[Certificate #0 CN=2k8r2-u-10-11
unable to get local issuer certificate]]></RESULTS>
<Status>Active</Status>
<First_Found_Datetime>2018-01-26T04:45:50Z</First_Found_Datetime>
<Last_Found_Datetime>2018-04-13T03:48:50Z</Last_Found_Datetime>
<Times_Found>111</Times_Found>
<Last_Test_Datetime>2018-04-13T03:48:50Z</Last_Test_Datetime>
<Last_Update_Datetime>2018-04-13T03:49:05Z</Last_Update_Datetime>
<Is_Ignored>0</Is_Ignored>
</Detection>
Sample - Host Detection XML Output, with truncation

A truncated response is returned when the API request returns more host records than the truncation limit. In this sample, the truncation limit is set to 100 host records.

API request:

```
curl -u "username:password" -H "X-Requested-With: curl" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=list&truncation_limit=100"
```
The Warning message in the XML output (shown below) indicates the URL you need to use to request the next 100 host records.

**XML output:**

```
... 
</DETECTION_LIST>
</HOST>
</HOST_LIST>
<WARNING>
  <CODE>1980</CODE>
  <TEXT>100 record limit exceeded. Use URL to get next batch of results.</TEXT>
</WARNING>
</RESPONSE>
</HOST_LIST_VM_DETECTION_OUTPUT>
```

**Sample - Filter superseded QIDs (filter_superseded_qids=1)**

In this example any QID superseded by another QID has been filtered out of the results. The XML output includes QID 370584 and QID 370613. QID 370610 was filtered out because it was superseded by QID 370613.

**API request:**

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=list&filter_superseded_qids=1"
```

**XML output:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_VM_DETECTION_OUTPUT SYSTEM "http://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/dtd/output.dtd">
<HOST_LIST_VM_DETECTION_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-06-03T10:22:34Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>1145</ID>
        <IP>10.10.10.9</IP>
        <TRACKING_METHOD>IP</TRACKING_METHOD>
        <OS><![CDATA[Windows 2003 Service Pack 2]]></OS>
        <DNS><![CDATA[win2003.sample.qualys.com]]></DNS>
        <DNS_DATA>
          <HOSTNAME>
```

544
<![[CDATA[win2003]]>  
</HOSTNAME>  
<DOMAIN>  
<![[CDATA[sample.qualys.com]]>  
</DOMAIN>  
<FQDN>  
<![[CDATA[win2003.sample.qualys.com]]>  
</FQDN>  
</DNS_DATA>  
<NETBIOS><![CDATA[LWIN2003HP1]]></NETBIOS>  
<LAST_SCAN_DATETIME>2018-01-08T19:50:18Z</LAST_SCAN_DATETIME>  
<LAST_VM_SCANNED_DATE>2018-01-08T19:36:29Z</LAST_VM_SCANNED_DATE>  
<LAST_VM_SCANNED_DURATION>619</LAST_VM_SCANNED_DURATION>  
<LAST_PC_SCANNED_DATE>2017-11-15T16:58:16Z</LAST_PC_SCANNED_DATE>  
<DETECTION_LIST>  
<DETECTION>  
<QID>370584</QID>  
<UNIQUE_VULN_ID>45654</UNIQUE_VULN_ID>  
<TYPE>Confirmed</TYPE>  
<SEVERITY>5</SEVERITY>  
<SSL>0</SSL>  
<RESULTS><![CDATA[C:\Program Files\Mozilla Firefox\firefox.exe Version is 42.0.0.0]]></RESULTS>  
<STATUS>Active</STATUS>  
-FIRST_FOUND_DATETIME>2017-10-10T10:30:48Z</FIRST_FOUND_DATETIME>  
-LAST_FOUND_DATETIME>2020-04-27T23:04:10Z</LAST_FOUND_DATETIME>  
<TIMES_FOUND>183</TIMES_FOUND>  
-LAST_TEST_DATETIME>2020-04-27T23:04:10Z</LAST_TEST_DATETIME>  
-LAST_UPDATE_DATETIME>2020-04-27T23:05:41Z</LAST_UPDATE_DATETIME>  
-IS_IGNORED>0</IS_IGNORED>  
-IS_DISABLED>0</IS_DISABLED>  
-LAST_PROCESSED_DATETIME>2020-04-27T23:05:41Z</LAST_PROCESSED_DATETIME>  
</DETECTION>  
<DETECTION>  
<QID>370613</QID>  
<UNIQUE_VULN_ID>43664</UNIQUE_VULN_ID>  
<TYPE>Confirmed</TYPE>  
<SEVERITY>5</SEVERITY>  
<SSL>0</SSL>  
<RESULTS><![CDATA[C:\Program Files\Google\Chrome\Application\33.0.1750.149\chrome.dll file version is 33.0.1750.149 %ProgramFiles%\Google\Chrome\Application\33.0.1750.149\chrome.dll file version is 33.0.1750.149]]></RESULTS>  
<STATUS>Active</STATUS>  
-FIRST_FOUND_DATETIME>2017-11-12T20:11:32Z</FIRST_FOUND_DATETIME>  
-LAST_FOUND_DATETIME>2020-04-
27T23:04:10Z</LAST FOUND DATETIME>
   <TIMES FOUND>162</TIMES FOUND>
   <LAST TEST DATETIME>2020-04-27T23:04:10Z</LAST TEST DATETIME>
   <LAST UPDATE DATETIME>2020-04-27T23:05:41Z</LAST UPDATE DATETIME>
   <IS IGNORED>0</IS IGNORED>
   <IS DISABLED>0</IS DISABLED>
   <LAST PROCESSED DATETIME>2020-04-27T23:05:41Z</LAST PROCESSED DATETIME>
</DETECTION>
</DETECTION_LIST>
</HOST>
</HOST_LIST>
</RESPONSE>
</HOST_LIST_VM_DETECTION_OUTPUT>

Sample - List assets with Qualys Detection Score (QDS)

**API request:**

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=list&ips=10.20.30.40,10.11.12.13&show_qds=1&qds_min=1&qds_max=20&show_qds_factors=1"

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_VM_DETECTION_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/dtd/output.dtd">
<HOST_LIST_VM_DETECTION_OUTPUT>
   <RESPONSE>
   <DATETIME>2022-01-31T12:10:01Z</DATETIME>
   <HOST_LIST> ...
   </HOST_LIST>
   <DETECTION>
   <QID>38170</QID>
   <UNIQUE_VULN_ID>52664</UNIQUE_VULN_ID>
   <TYPE>Confirmed</TYPE>
   <SEVERITY>2</SEVERITY>
   <PORT>443</PORT>
   <PROTOCOL>TCP</PROTOCOL>
   <SSL>1</SSL>
   <RESULTS>
   <![CDATA[Certificate #0
   CN=IPMI,OU=Software,O=Super_Micro_Computer,ST=California,C=US (IPMI)
doesn't resolve]]>}
   </RESULTS>
   <STATUS>ACTIVE</STATUS>
   <FIRST FOUND DATETIME>2021-12-29T14:09:58Z</FIRST FOUND DATETIME>
   <LAST FOUND DATETIME>2022-01-31T12:10:01Z</LAST FOUND DATETIME>
   </DETECTION>
</RESPONSE>
</HOST_LIST_VM_DETECTION_OUTPUT>
```
11T13:11:20Z</LAST_FOUND_DATETIME>

<QDS severity="LOW">5</QDS>

<QDS_FACTORS>
  <QDS_FACTOR name="RTI">
    <![CDATA[[No_Patch]]]>
  </QDS_FACTOR>
  <QDS_FACTOR name="TEMPORAL_SCORE">
    <![CDATA[2.1]]>
  </QDS_FACTOR>
  <QDS_FACTOR name="BASE_SCORE">
    <![CDATA[2.6]]>
  </QDS_FACTOR>
  <QDS_FACTOR name="SEVERITY">
    <![CDATA[2]]>
  </QDS_FACTOR>
  <QDS_FACTOR name="EXPLOIT_MATURITY">
    <![CDATA[null]]>
  </QDS_FACTOR>
  <QDS_FACTOR name="EXPLOIT_AVAILABLE">
    <![CDATA[[poc]]]>
  </QDS_FACTOR>
  <QDS_FACTOR name="TRENDING">
    <![CDATA[null]]>
  </QDS_FACTOR>
  <QDS_FACTOR name="MITIGATION_CONTROLS">
    <![CDATA[null]]>
  </QDS_FACTOR>
  <QDS_FACTOR name="MALWARE_NAME">
    <![CDATA[null]]>
  </QDS_FACTOR>
  <QDS_FACTOR name="MALWARE_HASH">
    <![CDATA[null]]>
  </QDS_FACTOR>
</QDS_FACTORS>

<TIMES_FOUND>1</TIMES_FOUND>

<LAST_TEST_DATETIME>2021-06-03T11:18:57Z</LAST_TEST_DATETIME>

<LAST_UPDATE_DATETIME>2021-06-05T03:12:47Z</LAST_UPDATE_DATETIME>

<IS_IGNORED>0</IS_IGNORED>

<IS_DISABLED>0</IS_DISABLED>

<LAST_PROCESSED_DATETIME>2021-06-05T03:12:47Z</LAST_PROCESSED_DATETIME>

...
Host List Detection - Normalized Data

Qualys normalizes the vulnerability scan results into the database using a complex and sophisticated process. This mechanism generates what is called the vulnerability "host based" scan results. Normalized data brings a lot of value to customers because they provide the latest complete vulnerability status for the hosts (NEW, ACTIVE, FIXED, REOPENED) and history information. Normalized data is completely independent of scan results and option profiles, as shown in the diagram below.

The Qualys database stores automatic data for VM scanned hosts. For each of these hosts there can be multiple detection records.

What is a VM Scanned Host? A VM scanned host is a host that has been successfully scanned by the Qualys VM service for vulnerabilities. Note that a host is considered successfully scanned when it was included as a scan target, the scan was launched and it completed successfully.

What is a Detection Record? A detection record is a unique instance of a discovered vulnerability for a given host. It identifies the host IP address, QID, port, service, FQDN and SSL flag (whether the vulnerability was detected over SSL).

Host List Detection - Use Cases

The host detection API is often used in conjunction with other information that can be downloaded using other Qualys APIs.
Create Custom Technical Reports with vulnerability details

Technical reports need additional information for each vulnerability such as the description, solution, threat or impact. The detection API provides the QID for each vulnerability found for an asset. The QID is a unique ID that references a vulnerability within the Qualys KnowledgeBase.

Use the following workflow to create custom technical reports:

Step 1 - Use the host list detection API to return “host based” vulnerability data for hosts in your account.

Step 2 - Use the KnowlegeBase API (/api/2.0/fo/knowledge_base/vuln/?action=list) to obtain vulnerability data, such as the vulnerability description, threat and impact. It’s possible to make a request for all vulnerabilities (QIDs) in the KnowledgeBase or just a specific vulnerability.

For example, to make a request for QID 90082 use the following URL:

https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/?action=list&ids=90082

where “qualysapi.qualys.com” is the name of the API server where your account is located (in this case US Platform 1).

Step 3 - Correlate the vulnerability information in the third party application using the QID number provided in the <QID> XML output which is returned by the host detection API (Step 1) and the KnowledgeBase API (Step 2).

A typical integration would be to create tables in a database for the XML output from both Qualys API functions and use QID as a key for a join. This way it would be possible to create queries that will provide all the vulnerabilities for a given set of hosts (according to custom search criteria) and their descriptions.

Get All PCI Vulnerabilities

Step 1 - First you need to create a dynamic search list titled “PCI Vulns” using the Qualys user interface. When creating the dynamic search list, select the PCI option next to Compliance Type as shown below.

![PCI Vulns search list](image)

Step 2 - Create an asset group titled “PCI Hosts” containing the hosts which are in scope for PCI compliance.

Step 3 - Make the following host list detection API request using the asset group title “PCI Hosts” and the search list title “PCI Vulns”:

https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?a
where "qualysapi.qualys.com" is the name of the API server where your account is located (in this case US Platform 1).

Host List Detection - Best Practices

Some background
When API calls are done to pull large sets of data, the backend will process data by streaming that information in batches to ensure data integrity and preventing overloading the backend services. That means that there will be brief periods of speeds declining while the next batch is being retrieved and processed to stream back to the client. However, the overall speed averages itself out in the long run.

You also need to keep in mind the contributing factors that could impact performance on a shared resource. Such as performing data pulls during peak usage, which will hit congestion and speeds will not be as fast as those conducted during off peak hours. There are also additional factors from the use of optional parameters used in API calls that do extra processing before streaming the data, active_kernels_only being an example.

Multi-Threading
We have been, and will continue to innovate and re-architect the capabilities of processing large amount of encrypted data for streaming through API to scale to our customers needs. While being able to provide customers with all of their Vulnerability information as quickly as possible is a primary focal point, it should be innovated in such a way that keeps data integrity in the forefront of every release. To do this, it takes time, effort, and dedicated resources to ensure full testing is done to account for all aspects. With that in mind, the use of automation, threading, and parallelism are techniques to that can assist with increasing performance with data pulls.

While fetching host information in an automated fashion, you can make use of multi-threading to collect data in batch sizes for optimum performance.

Maximum benefit has seen when the batch size is set evenly throughout the number of parallel threads used. For example, a host detection call resulting in a return of 100k assets, and using 10 threads in parallel, would benefit the most by using a batch size of \((100,000 / 10) = 10,000\). To reduce having one thread slow down the entire process by hitting a congested server, you can break this out further into batches of 5,000 hosts, resulting in 20 output files.

Looking for help? Check our examples here
Qualys API - Host List Detection API samples - Multithreading (GitHub)

Excluded Host List
/api/2.0/fo/asset/excluded_ip/?action=list
[GET] [POST]

Show the excluded host list for the user's account. Hosts in your excluded host list will not be scanned.

Permissions - Managers, Auditors view all excluded hosts in subscription. Unit Managers view excluded hosts in their own business unit. Scanners, Readers view excluded hosts in their account.

Express Lite - This API is available to Express Lite users.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td><em>(Required)</em></td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ips={value}</td>
<td><em>(Optional)</em> Show only certain excluded IP addresses/ranges. When unspecified, all excluded IPs/ranges in your account will be listed. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.24.1-10.10.24.20).</td>
</tr>
<tr>
<td>network_id={value}</td>
<td><em>(Optional and valid only when the Network Support feature is enabled for the user’s account)</em> Restrict the request to a certain custom network ID. You might need to use this parameter to get the excluded host list you’re interested in. See User Scenarios to know more about the behavior of this parameter.</td>
</tr>
<tr>
<td>ag_ids={value}</td>
<td><em>(Optional and valid only when the Network Support feature is enabled for the user’s account)</em> Restrict the request to a certain custom network ID. You might need to use this parameter to get the excluded host list you’re interested in.</td>
</tr>
<tr>
<td>ag_titles={value}</td>
<td><em>(Optional)</em> Show excluded hosts belonging to asset groups with certain strings in the asset group title. One or more asset group titles may be specified. Multiple entries are comma separated (for example, My+First+Asset+Group,Another+Asset+Group). These parameters are mutually exclusive and cannot be specified together: ag_ids and ag_titles.</td>
</tr>
<tr>
<td>use_tags={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
Let us consider different user scenarios to know more about the behavior of network_id parameter:

<table>
<thead>
<tr>
<th>User</th>
<th>Networks with access</th>
<th>network_id mandatory?</th>
<th>What does output include?</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>Global Default Network, Network 1, Network 2</td>
<td>No</td>
<td>Excluded host list from all the networks the user has access to.</td>
</tr>
<tr>
<td>User 2</td>
<td>Global Default Network</td>
<td>No</td>
<td>Excluded host list for global default network.</td>
</tr>
<tr>
<td>User 3</td>
<td>Network 1</td>
<td>Yes</td>
<td>Excluded host list for Network 1.</td>
</tr>
<tr>
<td>User 4</td>
<td>Network 1, Network 2, Network 3</td>
<td>Yes</td>
<td>Excluded host list for network that is listed in the request. Multiple entries are comma separated (for example, Network+1,Network+2,Network+3).</td>
</tr>
</tbody>
</table>

### Sample - List all excluded hosts

**API request:**

```
curl -u user:password -H "X-Requested-With: curl demo 2" -D headers.15 "https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/?action
```
Assets
Excluded Hosts Change History

=\text{list}\$

\textbf{XML output}

\begin{verbatim}
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE IP_LIST_OUTPUT SYSTEM
  "https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/ip_list
  _output.dtd">
<IP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-01-23T00:33:24Z</DATETIME>
    <IP_SET>
      <IP_RANGE network_id="0" expiration_date="2015-04-
      28T00:00:00Z">10.100.100.101-10.100.100.255</IP_RANGE>
      <IP network_id="14665885">10.10.10.1</IP>
      <IP network_id="0">10.100.100.100</IP>
    </IP_SET>
  </RESPONSE>
</IP_LIST_OUTPUT>
\end{verbatim}

\textbf{Sample - List all excluded hosts in IP range}

\textbf{API request:}

\begin{verbatim}
curl -u user:password -H "X-Requested-With: curl demo 2" -D
  headers.16
  "https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/
  ?action=list&ips=10.10.24.1-10.10.24.255"
\end{verbatim}

\textbf{DTD}

<platform API server>/api/2.0/fo/asset/excluded_ip/ip_list_output.dtd

\textbf{Excluded Hosts Change History}

\texttt{/api/2.0/fo/asset/excluded_ip/history/?action=list}

[GET] [POST]

View change history for excluded hosts in the user’s subscription. History record IDs in the
XML output are listed in decreasing order.

Permissions - Users with these roles have permission to view all excluded hosts in the
subscription: Manager, Auditor, Unit Manager, Scanner and Reader.

Unlike other APIs, an excluded hosts change history request returns change history
records for all relevant IP addresses in the subscription, regardless of whether the user has
access to these IP addresses in their account.
Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional) Show only certain excluded IP addresses/ranges. When unspecified, all excluded IPs/ranges in your subscription will be listed. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.24.1-10.10.24.20).</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional and valid only when the Network Support feature is enabled for the user’s account) Specify a network ID to restrict the request to a certain custom network.</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(Optional) Show only those history records in your subscription that have an ID number greater than or equal to an ID number you specify.</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(Optional) Show only those history records in your subscription that have an ID number less than or equal to an ID number you specify.</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Optional) Show only those history records in your subscription that have ID numbers matching the ID numbers you specify.</td>
</tr>
</tbody>
</table>

Sample - Change list for all excluded IPs

API request:
```
curl -u user:password -H "X-Requested-With: curl demo 2" -D
"headers.15
"https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/history
/?action=list"
```

XML output:
```
<!DOCTYPE HISTORY_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/history
/history_list_output.dtd">

<HISTORY_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-01-18T01:48:42Z</DATETIME>
    <HISTORY_LIST>
      <HISTORY>
        <ID>1923</ID>
        <IP_SET>
          <IP_RANGE>10.10.10.2-10.10.10.11</IP_RANGE>
        </IP_SET>
      </HISTORY>
    </HISTORY_LIST>
  </RESPONSE>
</HISTORY_LIST_OUTPUT>
```
<IP_RANGE>10.10.10.32-10.10.10.34</IP_RANGE>
<IP>10.10.30.70</IP>
</IP_SET>
ACTION>Added</ACTION>
<DATETIME>2017-12-02T05:19:06Z</DATETIME>
 USER_LOGIN>quays_ab</USER_LOGIN>
<COMMENTS><![CDATA[DD]]></COMMENTS>
</HISTORY>

<HISTORY>
 <ID>1863</ID>
 <IP_SET>
  <IP_RANGE>10.10.10.102-10.10.10.120</IP_RANGE>
 </IP_SET>
 ACTION>Removed</ACTION>
 <DATETIME>2017-06-01T23:51:26Z</DATETIME>
 USER_LOGIN>quays_ab</USER_LOGIN>
<COMMENTS><![CDATA[Removing 10.10.10.102-10.10.10.120]]></COMMENTS>
</HISTORY>

<HISTORY>
 <ID>1663</ID>
 <IP_SET>
  <IP_RANGE>10.10.10.100-10.10.10.120</IP_RANGE>
 </IP_SET>
 ACTION>Added</ACTION>
 <DATETIME>2016-04-29T06:56:13Z</DATETIME>
 USER_LOGIN>quays_ss</USER_LOGIN>
<COMMENTS><![CDATA[Scanner shouldn't add Exclude hosts]]></COMMENTS>
</HISTORY>

...

</HISTORY_LIST>
<WARNING>
 <CODE>1980</CODE>
 <TEXT>1,000 record limit exceeded. Use URL to get next batch of results.</TEXT>
<URL><![CDATA[https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/history/?action=list&id_max=1660]]></URL>
</WARNING>

<GLOSSARY>
 <USER_LIST>
  <USER>
   <USER_LOGIN>quays_ss</USER_LOGIN>
   <FIRST_NAME>Sally Unassigned</FIRST_NAME>
  </USER>
 </USER_LIST>
</GLOSSARY>
<LAST_NAME>Storm</LAST_NAME>  
<ROLE>Scanner</ROLE>  
</USER>  
</USER>  
<USER>  
<User_LOGIN>quays_ab</User_LOGIN>  
<FIRST_NAME>Al</FIRST_NAME>  
<LAST_NAME>Berger</LAST_NAME>  
<ROLE>Manager</ROLE>  
</USER>  
</USER_LIST>  
</GLOSSARY>  
</RESPONSE>  
</HISTORY_LIST_OUTPUT>

**DTD**

<platform API server>/api/2.0/fo/asset/excluded_ip/history/history_list_output.dtd

**Manage Excluded Hosts**

The excluded hosts endpoint (/api/2.0/fo/asset/excluded_ip) allows you to add and remove excluded hosts from your account.

**Add excluded hosts**

/api/2.0/fo/asset/excluded_ip/?action=add

[POST]

Add hosts (IPs) to your excluded host list. Hosts in your excluded host list will not be scanned.

Permissions - Managers and Unit Managers have permission to add IPs to the excluded host list.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=add</td>
<td>(Required)</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required) The IP addresses to be added to the excluded IPs list. Enter a comma separated list of IPv4 singletons or ranges. For example: 10.10.10.13,10.10.10.25-10.10.10.29</td>
</tr>
</tbody>
</table>
Sample - Add excluded hosts

API request:

```bash
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -d "action=add&ips=10.100.100.101-10.100.100.255&comment=adding ips&expiry_days=5" "https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/"
```

XML output:

```xml
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
 <RESPONSE>
  <DATETIME>2018-04-23T00:33:21Z</DATETIME>
  <TEXT>Adding IPs to Excluded IPs list.</TEXT>
  <ITEM_LIST>
   <ITEM>
    <KEY>Added IPs</KEY>
    <VALUE>10.100.100.101-10.100.100.255</VALUE>
   </ITEM>
  </ITEM_LIST>
 </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Add IPs already in excluded hosts list

**API request:**
```bash
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -d "action=add&ips=10.10.34.210-10.10.34.212&comment=adding, added IPs" "https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/"
```

**XML output:**
```xml
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-05-14T13:09:03Z</DATETIME>
    <TEXT>Not Adding any IPs to Excluded IPs list.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>IPs already in Excluded IPs list.</KEY>
        <VALUE>10.10.34.210-10.10.34.212</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Remove excluded hosts

/apif/2.0/fo/asset/excluded_ip/?action=remove

[POST]

Remove certain hosts from your excluded hosts list. You can choose to remove certain hosts (IPs) or all hosts from your excluded hosts list.

Permissions - Managers and Unit Managers have permission to remove IPs from the excluded host list.
Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=remove</td>
<td>(Required) The IP addresses to be removed from the excluded IPs list. Enter a comma separated list of IPv4 singletons or ranges. For example: 10.10.10.13,10.10.10.25-10.10.10.29</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Required) User-defined notes (up to 1024 characters).</td>
</tr>
<tr>
<td>comment={value}</td>
<td>(Optional and valid only when the user making the request has access to more than one network) Identify a network ID that is assigned to the IPs being removed from the excluded IPs list. By default, the user's default network ID is assigned.</td>
</tr>
</tbody>
</table>

Sample - Remove certain excluded hosts

**API request:**
```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -d "action=remove&ips=10.10.34.250-10.10.34.254&comment=remove IPS" "https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/"
```

**XML output:**
```
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-15T04:05:04Z</DATETIME>
    <TEXT>Removed IPs from Excluded IPs list.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>Removed IPs</KEY>
        <VALUE>10.10.34.250-10.10.34.254</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Remove all excluded hosts

/\api/2.0/fo/asset/excluded_ip/?action=remove_all

[POST]

Remove all hosts from your excluded hosts list.

Permissions - Managers and Unit Managers have permission to remove IPs from the excluded host list.
Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=remove_all</td>
<td>(Required)</td>
</tr>
<tr>
<td>comment=[value]</td>
<td>(Required) User-defined notes (up to 1024 characters).</td>
</tr>
<tr>
<td>network_id=[value]</td>
<td>(Optional and valid only when the user making the request has access to more than one network) Identify a network ID that is assigned to the IPs being removed from the excluded IPs list. By default, the user’s default network ID is assigned.</td>
</tr>
</tbody>
</table>

Sample - Remove all excluded hosts

**API request:**
```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -d "action=remove_all&comment=remove all ips" "https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/"
```

**XML output:**
```
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-24T00:08:19Z</DATETIME>
    <TEXT>Removed IPs from Excluded IPs list.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>Removed IPs</KEY>
        <VALUE>10.100.100.101-10.100.100.255,100.100.100.101-100.100.100.255</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

**DTD**
DTD returned by requests to add and remove excluded hosts
```
<platform API server>/api/2.0/simple_return.dtd
```

**Virtual Host List**

/api/2.0/fo/asset/vhost/?action=list

[GET] [POST]
List virtual hosts in the user's account. By default, all virtual hosts in the user's account are included.

Permissions - Managers view virtual hosts in the subscription. Unit Managers view virtual hosts in their own business unit. Scanners and Readers view virtual hosts in their own account.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=[0</td>
<td>1]</td>
</tr>
<tr>
<td>ip={value}</td>
<td>(Optional) Show only virtual hosts that have a certain IP address.</td>
</tr>
<tr>
<td>port={value}</td>
<td>(Optional) Show only virtual hosts that have a certain port.</td>
</tr>
</tbody>
</table>

**Sample - List virtual hosts in account**

**API request:**

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST "https://qualysapi.qualys.com/api/2.0/fo/asset/vhost/?action=list"
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE VIRTUAL_HOST_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/asset/vhost/vhost_list_output.dtd">
<VIRTUAL_HOST_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-04-26T11:20:42Z</DATETIME>
    <VIRTUAL_HOST_LIST>
      <VIRTUAL_HOST>
        <IP>10.11.65.3</IP>
        <PORT>255</PORT>
        <FQDN>asadfsadf-123.com</FQDN>
      </VIRTUAL_HOST>
      <VIRTUAL_HOST>
        <IP>10.11.65.5</IP>
        <PORT>246</PORT>
        <FQDN>asdfsahydk.com</FQDN>
      </VIRTUAL_HOST>
    </VIRTUAL_HOST_LIST>
  </RESPONSE>
</VIRTUAL_HOST_LIST_OUTPUT>
```
Manage Virtual Hosts

/api/2.0/fo/asset/vhost/?action={value}

[POST]

Create, edit and delete virtual hosts in the user account. One subscription can have a maximum of 5000 virtual hosts. The POST access method may be used to make an API request.

Permissions - Managers manage virtual hosts in the subscription. Unit Managers manage virtual hosts in their own business unit when granted this permission. Scanners have permission to manage virtual hosts in their account when granted this permission. Readers, Auditors do not have permission to manage virtual hosts.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=[action]</td>
<td>(Required) A flag used to make a virtual host request:</td>
</tr>
<tr>
<td></td>
<td>create (create a virtual host)</td>
</tr>
<tr>
<td></td>
<td>update (update/edit a virtual host)</td>
</tr>
<tr>
<td></td>
<td>delete (delete a virtual host)</td>
</tr>
<tr>
<td></td>
<td>add_fqdn (add one or more FQDNs to a virtual host)</td>
</tr>
<tr>
<td></td>
<td>delete_fqdn (remove one or more FQDNs from a virtual host)</td>
</tr>
<tr>
<td>echo_request=0/1</td>
<td>(Optional) Specify 1 to view (echo) input parameters in the XML output. By default these are not included.</td>
</tr>
<tr>
<td>ip=[value]</td>
<td>(Required) An IP address for the virtual host configuration.</td>
</tr>
<tr>
<td>network_id=[value]</td>
<td>(Optional) Network support must be enabled to specify the network_id. If network support is enabled and you do not provide a network_id, then the Default Global Network is considered. You can specify only one network_id.</td>
</tr>
<tr>
<td>port=[value]</td>
<td>(Required) A port number for the virtual host configuration.</td>
</tr>
<tr>
<td>fqdn=[value]</td>
<td>(Required for all actions except “delete”. Invalid for “delete”. )</td>
</tr>
<tr>
<td></td>
<td>One or more fully-qualified domain names (FQDNs) for the virtual host configuration. Multiple entries are comma separated.*</td>
</tr>
</tbody>
</table>
Sample - Create virtual host

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST
"action=create&ip=10.10.25.212&port=80&fqdn=www.abc123abc.com"
"https://qualysapi.qualys.com/api/2.0/fo/asset/vhost/"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2018-04-27T08:45:22Z</DATETIME>
        <TEXT>Virtual host successfully created.</TEXT>
    </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Create virtual host in a network

Specify network_id to create a virtual host in the specified network.

API request:
```
curl -u "username:password" -H "X-Requested-With: curl" -H
"Content-type: text/xml" -X POST
-d "action=create&network_id=5004&ip=10.10.10.20
&port=8080&fqdn=example1.fqdn.com,example2.fqdn.com"
"https://qualysapi.qualys.com/api/2.0/fo/asset/vhost/"
```

XML output:
```
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2019-11-22T07:27:52Z</DATETIME>
        <TEXT>Virtual host successfully created.</TEXT>
    </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Add FQDNs to a virtual host

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -H "X-
Requested-With:curl" -X POST
"action=add_fqdn&ip=10.10.25.212&port=80&fqdn=www.abc123abc.com"
"https://qualysapi.qualys.com/api/2.0/fo/asset/vhost/"
```

XML output:
```
```
<xml version="1.0" encoding="UTF-8" >
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-04-27T08:45:48Z</DATETIME>
    <TEXT>Virtual host FQDN(s) successfully added.</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>

More Samples
Qualys API - Virtual Host samples - Manage Virtual Hosts (GitHub)

DTD
<platform API server>/api/2.0/simple_return.dtd

Restricted IPs List
/api/2.0/fo/setup/restricted_ips/?action=list
[GET] [POST]

List restricted IPs within the user's subscription. Managers only have permission to perform these actions using this API.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=[0</td>
<td>1]</td>
</tr>
<tr>
<td>output_format={CSV</td>
<td>XML}</td>
</tr>
</tbody>
</table>

Sample - Download restricted IPs
API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=list"
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/" > output.txt

XML output:
The DTD for the restricted IPs list XML is provided in Appendix B - Ports used for scanning.
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE RESTRICTED_IPS_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/restricted_ips_output.dtd">
<RESTRICTED_IPS_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-03-22T11:12:56Z</DATETIME>
    <IP_SET>
      <IP_RANGE>10.10.10.1-10.10.10.255</IP_RANGE>
    </IP_SET>
    <STATUS>disabled</STATUS>
  </RESPONSE>
</RESTRICTED_IPS_OUTPUT>

DTD for restricted IPs list
<platform API server>/api/2.0/fo/setup/restricted_ips/restricted_ips_output.dtd

Sample - Download Restricted IPs List in CSV format

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=list&output_format=csv"
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/"

CSV output:
-----BEGIN_RESPONSE_BODY_CSV
10.0.0.0
10.0.0.101-10.255.255.255
-----END_RESPONSE_BODY_CSV
-----BEGIN_RESPONSE_FOOTER_CSV
STATUS
enabled
-----END_RESPONSE_FOOTER_CSV

Manage Restricted IPs

/api/2.0/fo/setup/restricted_ips/

[GET] [POST]

Manage and update the list of restricted IPs within the user's subscription. Managers only have permission to perform these actions using this API.
### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action={value}</td>
<td>(Required) The action for the request, one of: activate - enable or disable the restricted IPs feature clear - clear all restricted IPs and de-active this feature add - add restricted IPs delete - delete restricted IPs replace - replace restricted IPs</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>enable={0</td>
<td>1}</td>
</tr>
<tr>
<td>ips={value} -or- {CSV raw data upload}</td>
<td>(Optional and valid when action is add, replace or delete) The hosts you want to add to, remove from or replace in the restricted IPs list. IPs must be specified by using the &quot;ips&quot; parameter (using the POST method) or by uploading CSV raw data (using the GET or POST method). To upload CSV raw data using POST, specify --data-binary &lt;data&gt;. How to specify IP addresses. One or more IPs/ranges may be specified. Multiple IPs/ranges are comma separated. An IP range is specified with a hyphen (for example, 10.10.30.1-10.10.30.50). CIDR notation is supported.</td>
</tr>
</tbody>
</table>

### Sample - Replace restricted IPs

**API request:**

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=replace&ips=10.0.0.0/8" "https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/" > output.txt
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-03-22T11:45:00Z</DATETIME>
    <TEXT>Successfully replaced restricted ips</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>STATUS</KEY>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Delete restricted IPs, upload CSV raw data

CSV raw data:

```
$ cat file1.csv
10.0.0.1
10.0.0.2-10.0.0.100
```

API request:

```
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-03-22T11:45:34Z</DATETIME>
    <TEXT>Successfully deleted restricted ips</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>STATUS</KEY>
        <VALUE>disabled</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Activate Restricted IPs feature and enable list

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=activate&enable=1" "https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/" > output.txt
```
XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-03-22T11:46:45Z</DATETIME>
    <TEXT>Restricted IPs feature has been enabled successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>STATUS</KEY>
        <VALUE>enabled</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE.Return>

Sample - Clear All Restricted IPs and Disable the feature

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=clear"
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-03-22T12:04:34Z</DATETIME>
    <TEXT>Successfully cleared restricted ips</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>STATUS</KEY>
        <VALUE>disabled</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE.Return>
Asset Group List

/api/2.0/fo/asset/group/?action=list

[GET]  [POST]

List asset groups in the user’s account.

Permissions - Managers can view asset groups in the subscription. Unit Managers can view all asset groups in the user’s business unit (those assigned to the business unit, and those owned by all users in the business unit). Scanners and Readers can view asset groups in the user’s account (those assigned to the user, and those owned by the user).

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>output_format={csv</td>
<td>xml}</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Optional) Show only asset groups with certain IDs. Multiple IDs are comma separated.</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(Optional) Show only asset groups that have an ID greater than or equal to the specified ID.</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(Optional) Show only asset groups that have an ID less than or equal to the specified ID.</td>
</tr>
<tr>
<td>truncation_limit={value}</td>
<td>(Optional) Specify the maximum number of asset group records to output. By default this is set to 1000 records. If you specify truncation_limit=0, the output is not paginated and all records are returned in a single output. WARNING This can generate very large output and processing large XML files can consume a lot of resources on the client side. It is recommended to use the pagination logic and parallel processing. The previous page can be processed while the next page is being downloaded.</td>
</tr>
<tr>
<td>network_ids={value}</td>
<td>(Optional and valid only when the Networks feature is enabled in your account) Restrict the request to certain network IDs. Multiple IDs are comma separated.</td>
</tr>
<tr>
<td>unit_id={value}</td>
<td>(Optional) Show only asset groups that have a business unit ID equal to the specified ID.</td>
</tr>
<tr>
<td>user_id={value}</td>
<td>(Optional) Show only asset groups that have a user ID equal to the specified ID.</td>
</tr>
</tbody>
</table>
**Parameter** | **Description**
---|---
title={value} | (Optional) Show only the asset group that has a title equal to the specified string - this must be an exact match.
show_attributes={value} | (Optional) Show attributes for each asset group along with the ID. Specify ALL or a comma-separated list of attribute names. Attribute names: ID, TITLE, OWNER_USER_NAME, OWNER_USER_ID, OWNER_UNIT_ID, NETWORK_IDS, LAST_UPDATE, IP_SET, APPLIANCE_LIST, DOMAIN_LIST, DNS_LIST, NETBIOS_LIST, EC2_ID_LIST, HOST_IDS, ASSIGNED_USER_IDS, ASSIGNED_UNIT_IDS, BUSINESS_IMPACT, CVSS, COMMENTS.

**Sample - List asset groups, show default attributes**

**API request:**

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=list&ids=442838" "https://qualysapi.qualys.com/api/2.0/fo/asset/group/"
```

**XML output:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ASSET_GROUP_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/asset/group/asset_group_list_output.dtd">
<ASSET_GROUP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-05-17T08:48:41Z</DATETIME>
    <ASSET_GROUP_LIST>
      <ASSET_GROUP>
        <ID>442838</ID>
        <TITLE><![CDATA[All]]></TITLE>
        <OWNER_ID>103448</OWNER_ID>
        <UNIT_ID>0</UNIT_ID>
        <NETWORK_ID>0</NETWORK_ID>
        <IP_SET>
          <IP_RANGE>10.10.10.0-10.10.10.1</IP_RANGE>
          <IP_RANGE>10.10.10.3-10.10.10.6</IP_RANGE>
          <IP>10.10.10.14</IP>
          <IP_RANGE>10.10.10.16-10.10.10.20</IP_RANGE>
          <IP_RANGE>10.10.10.22-10.10.10.255</IP_RANGE>
          <IP>10.10.31.26</IP>
        </IP_SET>
      </ASSET_GROUP>
    </ASSET_GROUP_LIST>
  </RESPONSE>
</ASSET_GROUP_LIST_OUTPUT>
```
Sample - List asset groups, show all attributes

API request:
```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=list&ids=246385&show_attributes=ALL"
"https://qualysapi.qualys.com/api/2.0/fo/asset/group/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ASSET_GROUP_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/group/asset_group_list_output.dtd">
<ASSET_GROUP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-03-17T09:52:59Z</DATETIME>
    <ASSET_GROUP_LIST>
      <ASSET_GROUP>
        <ID>246385</ID>
        <TITLE>user_john</TITLE>
        <OWNER_USER_ID>180603</OWNER_USER_ID>
        <LAST_UPDATE>2018-03-07T11:37:57Z</LAST_UPDATE>
        <BUSINESS_IMPACT>High</BUSINESS_IMPACT>
        <DEFAULT_APPLIANCE_ID>199673</DEFAULT_APPLIANCE_ID>
        <APPLIANCE_IDS>199673, 199674</APPLIANCE_IDS>
        <IP_SET>
          <IP_RANGE>10.10.10.10-10.10.10.11</IP_RANGE>
          <IP_RANGE>10.113.197.131-10.113.197.132</IP_RANGE>
        </IP_SET>
        <DNS_LIST>
          <DNS>qualsss1.com</DNS>
        </DNS_LIST>
        <NETBIOS_LIST>
          <NETBIOS>WIN2003-SRV-O</NETBIOS>
        </NETBIOS_LIST>
        <HOST_IDS>634744, 653133</HOST_IDS>
        <ASSIGNED_USER_IDS>198400, 198401</ASSIGNED_USER_IDS>
        <ASSIGNED_UNIT_IDS>202741</ASSIGNED_UNIT_IDS>
        <OWNER_USER_NAME>John Doe</OWNER_USER_NAME>
      </ASSET_GROUP>
    </ASSET_GROUP_LIST>
  </RESPONSE>
</ASSET_GROUP_LIST_OUTPUT>
```

DTD for asset group list
```
<platform API server>/api/2.0/fo/asset/group/asset_group_list_output.dtd
```
Manage Asset Groups

Create, edit and delete asset groups in the user’s account.

Permissions - Managers can manage (create, edit, delete) all asset groups in the subscription. Unit Managers can manage asset groups owned by any user in the user’s same business unit. Scanners and Readers can manage asset groups owned by the user.

Add new asset group
/api/2.0/fo/asset/group/?action=add

[POST]

Add a new asset group in the user’s account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=add</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0</td>
<td>(Optional) Specify 1 to show (echo) the request’s input parameters (names, values) in the XML output. When unspecified, parameters are not included in the XML output.</td>
</tr>
<tr>
<td>title=value</td>
<td>(Required) An asset group title. This name must be unique and can’t be “All”.</td>
</tr>
<tr>
<td>network_id=value</td>
<td>(Optional) The network ID of the network you want to assign the asset group to.</td>
</tr>
</tbody>
</table>

Sample - Add asset group

API request:

curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "title=MY DEMO AG&network_id=1220&comments=This is this is division&location=this is location&business_impact=high&cvss_enviro_cdp=low&cvss_enviro_td=low&cvss_enviro_cr=medium&cvss_enviro_ir=high&cvss_enviro_ar=medium&ips=10.1.1.1/31" "https://qualysapi.qualys.com/api/2.0/fo/asset/group/?action=add"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
<DATETIME>2018-03-28T22:57:50Z</DATETIME>
<Text>Asset Group successfully added.</Text>
<item_list>
  <item>
    <key>ID</key>
    <value>395752377</value>
  </item>
</item_list>
</response>
</simple_return>
Edit asset group

/api/2.0/fo/asset/group/?action=edit

[POST]

Edit an existing asset group in the user’s account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=edit</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) The ID of the asset group you want to edit.</td>
</tr>
<tr>
<td>[parameters]</td>
<td>See “Asset Group Parameters”</td>
</tr>
</tbody>
</table>

Sample - Edit asset group

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "id=395752377&set_title=MY ASSET GROUP"
"https://qualysapi.qualys.com/api/2.0/fo/asset/group/?action=edit"

XML output:
The XML output uses the simple return (/api/2.0/simple_return.dtd).

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
<RESPONSE>
<DATETIME>2014-05-29T15:29:00Z</DATETIME>
<TEXT>Asset Group Updated Successfully</TEXT>
<ITEM_LIST>
<ITEM>
<KEY>ID</KEY>
<VALUE>395752377</VALUE>
</ITEM>
</ITEM_LIST>
</RESPONSE>
</SIMPLE_RETURN>
```
Delete asset group

/api/2.0/fo/asset/group/?action=delete

[POST]

Delete an asset group present in the user's account. By deleting an asset group any scheduled scans using the asset group will be deactivated.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=</td>
<td>Optional) Specify 1 to show (echo) the request's input parameters (names, values) in the XML output. When unspecified, parameters are not included in the XML output.</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) The ID of the asset group you want to delete.</td>
</tr>
</tbody>
</table>

Sample - Delete asset group

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "id=395752377" "https://qualysapi.qualys.com/api/2.0/fo/asset/group/?action=delete"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-03-29T15:49:35Z</DATETIME>
    <TEXT>Asset Group Deleted Successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>395752377</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Asset Group Parameters

Theses parameters are used for adding and editing an asset group.

The "set" (overwrite) and "remove" operations can cause the asset group to have no IPs, domains, etc depending on the parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Name</th>
<th>Parameter Name</th>
<th>Parameter Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>action=add</td>
<td>action=edit</td>
<td>action=edit</td>
</tr>
<tr>
<td>Comments</td>
<td>comments</td>
<td>set_comments</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(255 characters maximum)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td>division</td>
<td>set_division</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(64 characters maximum)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>function</td>
<td>set_function</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(64 characters maximum)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>location</td>
<td>set_location</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(64 characters maximum)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Impact</td>
<td>business_impact</td>
<td>set_business_impact</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(One of: critical, high, medium, low, none)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP addresses/ranges</td>
<td>ips</td>
<td>add_ips</td>
<td>remove_ips</td>
</tr>
<tr>
<td></td>
<td></td>
<td>set_ips</td>
<td></td>
</tr>
<tr>
<td>Scanner Appliances</td>
<td>appliance_ids</td>
<td>add_appliance_ids</td>
<td>remove_appliance_ids</td>
</tr>
<tr>
<td></td>
<td>(Looking for appliance IDs? Use the Appliance API (/api/2.0/fo/appliance/). See KnowledgeBase)</td>
<td>set_appliance_ids</td>
<td></td>
</tr>
<tr>
<td>Default Scanner Appliance</td>
<td>default_appliance_id</td>
<td>set_default_appliance_id</td>
<td></td>
</tr>
<tr>
<td>Domains</td>
<td>domains</td>
<td>add_domains</td>
<td>remove_domains</td>
</tr>
<tr>
<td></td>
<td></td>
<td>set_domains</td>
<td></td>
</tr>
<tr>
<td>DNS Names</td>
<td>dns_names</td>
<td>add_dns_names</td>
<td>remove_dns_names</td>
</tr>
<tr>
<td></td>
<td></td>
<td>set_dns_names</td>
<td></td>
</tr>
<tr>
<td>NetBIOS Names</td>
<td>netbios_names</td>
<td>add_netbios_names</td>
<td>remove_netbios_names</td>
</tr>
<tr>
<td></td>
<td></td>
<td>set_netbios_names</td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>title</td>
<td>set_title</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(255 characters maximum)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CVSS Environmental Metric: Collateral Damage Potential</td>
<td>cvss_enviro_cdp</td>
<td>set_cvss_enviro_cdp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(One of: high, medium-high, low-medium, low, none)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Purge Hosts

`/api/2.0/fo/asset/host/?action=purge`

[POST]

Purge hosts in your account to remove the assessment data associated with them.

Purging hosts will remove host based data in the user’s account (scan results will not be removed). Purged host information will not appear in new reports generated by users. One or both types of host data is removed, based on the user’s API request: vulnerability data and compliance data.

Permissions

Managers can purge assessment data for all hosts in the subscription, including vulnerability data and/or compliance data.

Auditors can purge compliance data only for all compliance hosts in the subscription (vulnerability data will not be removed).

Unit Managers, Scanners, and Readers can purge vulnerability data and/or compliance data in their user account if granted the permission “Purge host information/history”. The permission “Manage compliance” is required to purge compliance data.

Express Lite - This API is available to Express Lite users.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Name</th>
<th>Parameter Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVSS Environmental Metric:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target Distribution</td>
<td>cvss_enviro_td</td>
<td>set_cvss_enviro_td</td>
</tr>
<tr>
<td>(One of: high, medium, low, none)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CVSS Environmental Metric:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confidentiality Requirement</td>
<td>cvss_enviro_cr</td>
<td>set_cvss_enviro_cr</td>
</tr>
<tr>
<td>(One of: high, medium, low)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CVSS Environmental Metric:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrity Requirement</td>
<td>cvss_enviro_ir</td>
<td>set_cvss_enviro_ir</td>
</tr>
<tr>
<td>(One of: high, medium, low)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CVSS Environmental Metric:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Availability Requirement</td>
<td>cvss_enviro_ar</td>
<td>set_cvss_enviro_ar</td>
</tr>
<tr>
<td>(One of: high, medium, low)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to choose data scope for asset purge

The input parameter “data_scope” allows you to specify the type of data to purge from a host. Specify “vm” to purge vulnerability data, “pc” to purge compliance data, or “vm,pc” (irrespective of order) to purge both types of data.

You can also use the input parameter “compliance_enabled” to purge compliance data along with vulnerability data or vulnerability data only. This option does not allow you to purge compliance data only.

You can combine compliance_enabled and data_scope in the same request. Note, however, that anytime compliance_enabled=1 is specified, then both vulnerability and compliance data is purged regardless of the data_scope value. See the table below to understand the different combinations and the type of data purged.

<table>
<thead>
<tr>
<th>compliance_enabled value</th>
<th>data_scope value</th>
<th>type of data purged</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>unspecified</td>
<td>vulnerability + compliance data</td>
</tr>
<tr>
<td>0</td>
<td>unspecified</td>
<td>vulnerability data only</td>
</tr>
<tr>
<td>unspecified or 0</td>
<td>vm</td>
<td>vulnerability data only</td>
</tr>
<tr>
<td>unspecified or 0</td>
<td>pc</td>
<td>compliance data only</td>
</tr>
<tr>
<td>unspecified or 0</td>
<td>vm,pc</td>
<td>vulnerability + compliance data</td>
</tr>
<tr>
<td>1</td>
<td>vm</td>
<td>vulnerability + compliance data</td>
</tr>
<tr>
<td>1</td>
<td>pc</td>
<td>vulnerability + compliance data</td>
</tr>
<tr>
<td>1</td>
<td>vm,pc</td>
<td>vulnerability + compliance data</td>
</tr>
</tbody>
</table>

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=purge</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Optional) Purge host information for certain host IDs/ranges. One or more host IDs/ranges may be specified. Multiple entries are comma separated. A host ID range is specified with a hyphen (for example, 190-400). Valid host IDs are required. One of these host selection parameters must be specified in an API request: ids, ips, ag_ids or ag_titles. Multiple host selection parameters may be specified together in the same request.</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional) Purge host information certain IP addresses/ranges. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.10.1-10.10.10.100).</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ag_ids={value}</td>
<td>(Optional) Purge hosts belonging to asset groups with certain IDs. One or more asset group IDs and/or ranges may be specified. Multiple entries are comma separated. A range is specified with a dash (for example, 386941-386945). Valid asset group IDs are required. One of these host selection parameters must be specified in an API request: ids, ips, ag_ids or ag_titles. Multiple host selection parameters may be specified together in the same request.</td>
</tr>
<tr>
<td>ag_titles={value}</td>
<td>(Optional) Purge hosts belonging to asset groups with certain strings in the asset group title. One or more asset group titles may be specified. Multiple entries are comma separated (for example, My+First+Asset+Group,Another+Asset+Group). One of these parameters must be specified in an API request: ids, ips, ag_ids or ag_titles. Multiple host selection parameters may be specified together in the same request. These parameters are mutually exclusive and cannot be specified together: ag_ids and ag_titles.</td>
</tr>
<tr>
<td>network_ids={value}</td>
<td>(Optional, and valid only when the Network Support feature is enabled for the user’s account) Restrict the request to certain custom network IDs. Multiple network IDs are comma separated.</td>
</tr>
<tr>
<td>no_vm_scan_since={date}</td>
<td>(Optional) Purge hosts not scanned since a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”. User Permissions: An Auditor cannot be specify this parameter.</td>
</tr>
<tr>
<td>no_compliance_scan_since={date}</td>
<td>(Optional) Purge compliance hosts not scanned since a certain date and time (optional). This parameter is invalid for an Express Lite user. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”. User Permissions: A sub-account (Unit Manager, Scanner or Reader) can specify this parameter only when the user account is granted certain permissions to purge compliance information. See “Input Parameters”.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| data_scope={(value)}     | (Optional) The type of data to purge. Specify “vm” to purge vulnerability data, specify “pc” to purge compliance data, or specify both as a comma separated list to purge both types of data.  
If compliance_enabled=1 is specified in the same request, then vulnerability and compliance data will both be purged regardless of the data_scope value. |
| compliance_enabled={0|1} | (Optional) This parameter is valid only when the policy compliance module is enabled for the user account.  
Specify 1 to purge compliance hosts in the user’s account. These hosts are assigned to the PC module. When selected, the service will remove vulnerability data and compliance data associated with the selected hosts.  
Specify 0 to purge hosts which are not assigned to the PC module. When specified (without data_scope), the service will remove only vulnerability information associated with the selected hosts.  
Note: A sub-account (Unit Manager, Scanner or Reader) can specify this parameter only when the user account is granted permissions to purge compliance information. An Auditor does not have permission to set compliance_enabled=0. |
| os_pattern={(expression)} | (Optional) Purge only hosts which have an operating system matching a certain regular expression. An empty value cannot be specified. Use “%5E%24” to match empty string.  
Important: The regular expression string you enter must follow the PCRE standard and it must be URL encoded.  
Sample regular expression strings for matching OS names:  
Qualys API - Host List Detection API samples (GitHub, see sample 17)  
For information about the Perl Compatible Regular Expressions (PCRE) standard visit:  
For the PCRE syntax, see:  
Sample 1 - Purge only compliance data
In this example, data_scope=pc so only compliance data will be purged for the host.

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -X "POST" -d "action=purge&ips=10.20.32.152&data_scope=pc" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/
```

Response:
```
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-11-19T10:51:57Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Hosts Queued (compliance data) for Purging</TEXT>
        <ID_SET>
          <ID>3971339</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample 2 - Purge only vulnerability data
In this example, data_scope=vm so only vulnerability data will be purged for the host.

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -X "POST" -d "action=purge&ips=10.20.32.152&data_scope=vm" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/
```

Response:
```
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-11-19T10:51:45Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Hosts Queued (vulnerability data) for Purging</TEXT>
        <ID_SET>
          <ID>3971339</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```
Sample 3 - Purge vulnerability and compliance data
In this example, data_scope=pc,vm so both vulnerability and compliance data will be purged for the host.

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -X "POST" -d "action=purge&ips=10.20.32.152&data_scope=pc,vm" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/

Response:
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-11-19T10:52:12Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Hosts Queued (vulnerability + compliance data) for Purging</TEXT>
        <ID_SET>
          <ID>3971339</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>

Sample 4 - Purge vulnerability and compliance data (using compliance_enabled)
In this example, compliance_enabled=1 and data_scope=pc. Both vulnerability and compliance data will be purged for the host since compliance_enabled=1 takes precedence.

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -X "POST" -d "action=purge&ips=10.20.32.154&compliance_enabled=1&data_scope=vm" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/

Response:
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2020-11-19T10:52:12Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Hosts Queued (vulnerability + compliance data) for Purging</TEXT>
        <ID_SET>
          <ID>3971339</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
Sample 5 - Purge only vulnerability data (using compliance_enabled)

In this example, compliance_enabled=0 and data_scope=vm so only vulnerability data will be purged.

API request:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -X "POST" -d "action=purge&ips=10.20.32.154&compliance_enabled=0&data_scope=vm" 
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/
```

Response:

```xml
<DOCTYPE BATCH_RETURN SYSTEM 
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
<RESPONSE>
<DATETIME>2020-11-19T11:25:12Z</DATETIME>
<BATCH_LIST>
<BATCH>
<TEXT>Hosts Queued (vulnerability data) for Purging</TEXT>
<ID_SET>
<ID>3971340</ID>
</ID_SET>
</BATCH>
</BATCH_LIST>
</RESPONSE>
</BATCH_RETURN>
```

DTD

```xml
<platform API server>/api/2.0/fo/asset/host/dtd/purge/output.dtd
```
The Patch API lets you view the list of all superseding patches for detection on specific host. For the host, the Patch Info List provides information such as detection QID, patch QID, patch severity, patch title, patch vendor ID, patch release date, and patch links.

User permissions - Managers and Unit Managers can fetch the patch list on assets in their own business unit. Scanners and Readers fetch the patch list on assets in their own account.

### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_id={value}</td>
<td>(Required) The output lists all the superseding patches that will fix the detections on a single host instance. Specify the ID for the host to include in the report. A valid host ID must be entered.</td>
</tr>
<tr>
<td>output_format={xml}</td>
<td>(Optional) Specifies the format of the host detection list output. When not specified, the output format is xml. A valid value is xml.</td>
</tr>
</tbody>
</table>

### Sample 1: Patch List

**API request:**

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "X-Requested-With: curl"
-H "Content-Type: text/xml"
"host_id=136801&output_format=xml"
"https://qualysapi.qualys.com/api/2.0/fo/asset/patch/index.php"
```

**XML output:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE PATCH_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/asset/patch/host_patches.dtd">
<PATCH_LIST_OUTPUT>
  <RESPONSE>
    <SUBSCRIPTION_ID>3058</SUBSCRIPTION_ID>
    <HOST_ID>136801</HOST_ID>
    <IP>10.10.25.249</IP>
    <DNS><![CDATA[ora11107-25-249]]></DNS>
    <NETBIOS><![CDATA[ORA11107-25-249]]></NETBIOS>
    <OS><![CDATA[Windows 2003 Service Pack 2]]></OS>
    <OS_CPE><![CDATA[]]></OS_CPE>
  </RESPONSE>
</PATCH_LIST_OUTPUT>
```
<NETWORK><![CDATA[Star Trek]]></NETWORK>
<PATCH_INFO_LIST>
  <PATCH_INFO>
    <DETECTION_QIDS>
      <QID cve_ids=""><![CDATA[19883]]></QID>
    </DETECTION_QIDS>
    <PATCH_QID cve_ids=""><![CDATA[19883]]></PATCH_QID>
    <PATCH_SEVERITY>4</PATCH_SEVERITY>
    <PATCH_TITLE><![CDATA[Oracle 11.1.0.7 on Microsoft Windows - General Update Multiple Issues (Patch #54)]]></PATCH_TITLE>
    <PATCH_VENDOR_ID><![CDATA[11.1.0.7 Patch 54 - 32bit, 11.1.0.7 Patch 54 - 64bit]]></PATCH_VENDOR_ID>
    <PATCH_RELEASE_DATE>2013-10-15 00:00:00</PATCH_RELEASE_DATE>
    <PATCH_LINKS>
      <LINK os_sw="Windows"><![CDATA[https://support.oracle.com/epmos/faces/ui/patch/PatchDetail.jspx?patchId=17363759]]></LINK>
      <LINK os_sw="Windows"><![CDATA[https://support.oracle.com/epmos/faces/ui/patch/PatchDetail.jspx?patchId=17363760]]></LINK>
    </PATCH_LINKS>
  </PATCH_INFO>
</PATCH_INFO_LIST>

DTD
<platform API server>/api/2.0/fo/asset/patch/host_patches.dtd
IPv6 Assets

The IPv6 Assets API allows Manager users to manage IPv6 assets so they can be scanned using Qualys. The IPv6 API can be used when the IPv6 Support feature is enabled in the user's subscription. Please contact Support if you would like this feature enabled for your account.

API Support for IPv6 Asset Management and Scanning
IPv6 Mapping Record List
Add IPv6 Mapping Records

API Support for IPv6 Asset Management and Scanning
IPv6 Support is a subscription-level option that must be enabled for your subscription by Qualys Support in order to start managing and scanning IPv6 hosts. Follow the steps below to get started with managing and scanning IPv6 hosts using the API.

Step 1: Add Special IPv4 Addresses to your subscription

Using the Asset API add to your subscription the special, mapping IPv4 addresses. These IPv4 addresses are used for mapping IPv4 addresses to your IPv6 hosts. The IPv4 addresses for mapping are in the special 0.0.0.0/8 network, in this range:

0.0.0.1-0.254.255.255

A sample request for adding the special IPv4 addresses is shown below (where qualysapi.qualys.com is the server URL where your Qualys account is located):

https://qualysapi.qualys.com/msp/asset_ip.php?action=add&host_ips=0.0.0.1-0.0.0.255

Step 2: Add IPv6 Mapping Records

Manager users can add IPv6 mapping records for the subscription by submitting the records in CSV or XML format. Each mapping record associates one IPv6 address in your network to one IPv4 address in the special mapping range 0.0.0.1-0.254.255.255. A maximum of 10,000 records can be added or removed per API request.

How to Add IPv6 Records in CSV

Review the steps below to learn how to add IPv6 mapping records by submitting the records in CSV format. A curl client is used to illustrate this process.

1) View Mapping Records in CSV

API request:

```bash
$ curl -u username:password -H "X-Requested-With: curl" "https://qualysapi.qualys.com/api/2.0/fo/asset/ip/v4_v6/?action=list&output_format=csv"
```
XML output:

Note: The service automatically returns an ID value in the ID column for each IPv6 mapping record. This ID is assigned by the service when the record is created.

```
----BEGIN_RESPONSE_BODY_CSV
ID,IPv4,IPv6
"46947","0.0.0.7","2001:db8:85a3::8a2e:370:84"
"47036","0.0.0.1","2001:db8:85a3::8a2e:370:77"
----END_RESPONSE_BODY_CSV
----BEGIN_RESPONSE_FOOTER_CSV
"Status Message"
"Finished"
----END_RESPONSE_FOOTER_CSV
```

2) Prepare file1.csv with records to be added

The CSV file contents identify one or more IPv6 mapping records to be added. The columns in the CSV upload file are described below.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4</td>
<td>(Required) An IPv4 address. The IPv4 address can be defined in only one IPv6 mapping record within your subscription.</td>
</tr>
<tr>
<td>IPv6</td>
<td>(Required) An IPv6 address. The IPv6 address can be defined in only one IPv6 mapping record within your subscription.</td>
</tr>
<tr>
<td>ID</td>
<td>(Optional) A user-defined, custom ID may be included. Important: Custom ID values will not be saved with record data within your subscription.</td>
</tr>
</tbody>
</table>

The CSV file must include the input parameters action=add and csv_data=. The parameter all_or_nothing is optional. When set to 1 or unspecified, the service cancels the request and does not add any new records if it finds the upload data has one record with an IP conflict. When set to 0 the service does not cancel the request if an IP conflict is found.

Sample file1.csv used to add IPv6 mapping records:

```
$ cat file1.csv
action=add&all_or_nothing=1&csv_data=
"0.0.0.2","2001:470:8418:a18::a0a:1805"%A
"0.0.0.3","2001:470:8418:a18::a0a:ab7"%A
"0.0.0.4","2001:470:8418:a18::a0a:1849"%A
"0.0.0.5","2001:470:8418:a18::a0a:189c"%A
"0.0.0.6","2001:470:8418:a18::a0a:189d"%A
"0.0.0.8","2001:470:8418:a18::a0a:189e"%A
```
IPv6 Assets
API Support for IPv6 Asset Management and Scanning

"0.0.0.9"","2001:470:8418:a18::a0a:18d0"%0A"0.0.0.10","2001:470:8418:a18::a0a:18d1"%0A"0.0.0.11","2001:470:8418:a18::a0a:18d2"%0A"0.0.0.12","2001:470:8418:a18::a0a:18d6"%0A"0.0.0.13","2001:470:8418:a18::a0a:18d7"%0A"0.0.0.14","2001:470:8418:a18::a0a:18da"%0A"0.0.0.15","2001:470:8418:a18::a0a:18db"%0A"0.0.0.16","ff00:abcd::1234"%0A

3) POST data from file1.csv (Success)
Input:
$ curl -u username:password -H "X-Requested-With: curl"
   -d @file1.csv
   "https://qualysapi.qualys.com/api/2.0/fo/asset/ip/v4_v6/
Output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
  "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2011-11-03T19:31:27Z</DATETIME>
    <TEXT>Successfully imported 14 records</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>

How to Add IPv6 Records in XML
Review the steps below to learn how to add IPv6 mapping records by submitting the records in XML format. A curl client is used to illustrate this process.

1) View mapping records in XML
API request:
$ curl -u username:password -H "X-Requested-With: curl"
   "https://qualysapi.qualys.com/api/2.0/fo/asset/ip/v4_v6/?action=list&output_format=xml"
Output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE IP_MAP_LIST_OUTPUT SYSTEM
  "https://qualysapi.qualys.com/api/2.0/ip_map_list.dtd">
<IP_MAP_LIST_OUTPUT>
  <IP_MAP_LIST>
    <IP_MAP>
      <ID>002</ID>
      <IP>0.0.0.9</IP>
      <ASSET>1</ASSET>
      <GROUP>North America</GROUP>
      <NET_RANGE>2001:470:8418:a18::a0a:18d0</NET_RANGE>
    </IP_MAP>
    ...
  </IP_MAP_LIST>
</IP_MAP_LIST_OUTPUT>

Note: The service automatically returns an ID value in the <ID> element for each IPv6 mapping record. This ID is assigned by the service when the record is created.
IPv6 Assets
API Support for IPv6 Asset Management and Scanning

"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/v4_v6/ip_map_list_output.dtd">
<IP_MAP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2011-11-28T19:42:10Z</DATETIME>
    <IP_MAP_LIST>
      <IP_MAP>
        <ID>46947</ID>
        <V4>0.0.0.7</V4>
        <V6>2001:db8:85a3::8a2e:370:84</V6>
      </IP_MAP>
      <IP_MAP>
        <ID>47036</ID>
        <V4>0.0.0.1</V4>
        <V6>2001:db8:85a3::8a2e:370::77</V6>
      </IP_MAP>
    </IP_MAP_LIST>
  </RESPONSE>
</IP_MAP_LIST_OUTPUT>

2) Prepare file2.xml with records to be added
The XML file contents identify one or more IPv6 mapping records to be added. The element in the XML upload file are described below.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;V4&gt;</td>
<td>(Required) An IPv4 address. The IPv4 address can be defined in only one IPv6 mapping record within your subscription.</td>
</tr>
<tr>
<td>&lt;V6&gt;</td>
<td>(Required) An IPv6 address. The IPv6 address can be defined in only one IPv6 mapping record within your subscription.</td>
</tr>
<tr>
<td>&lt;ID&gt;</td>
<td>(Optional) A user-defined, custom ID may be included. Important: Custom ID values will not be saved with record data within your subscription.</td>
</tr>
</tbody>
</table>

The XML file must include the input parameters action=add and xml_data=. The parameter all_or_nothing is optional. When set to 1 or unspecified, the service cancels the request and does not add any new records if it finds the upload data has one record with an IP conflict. When set to 0 the service does not cancel the request if an IP conflict is found.

Sample file2.xml used to add IPv6 mapping records:

```
$ cat file2.xml
action=add&xml_data=
<IP_MAP_LIST>
```

585
IPv6 Assets
API Support for IPv6 Asset Management and Scanning

3) POST data from file2.xml (Success)

API request:
```
$ curl -u username:password -H "X-Requested-With: curl"
-d @file2.xml
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/v4_v6/
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2011-11-03T20:59:07Z</DATETIME>
    <TEXT>Successfully imported 2 records</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Step 3: Enable IPv6 for Scanner Appliance(s)
IPv6 scanning is supported using a scanner appliance enabled with IPv6. You can enable this by editing the appliance within the Qualys user interface. Once IPv6 is enabled, the appliance uses stateless address autoconfiguration to obtain an IPv6 address from the router (note that stateful configuration through DHCPv6 or Static IPv6 is not supported).

Step 4: Launch Scan
Using the Qualys API you can launch scans on the IPv4 addresses which are mapped to IPv6 addresses.

Step 5: View IPv6 Addresses using Host List Detection API
The scan results XML output will include IPv4 addresses only. Also, scan reports downloaded from the user interface will include IPv4 addresses only.

The host list detection output returned from a host list detection API request (api/2.0/fo/asset/host/vm/detection/?action=list ) gives you the IPv6 address, if available, along with the "automatic" vulnerability detection data.
To request a list of VM scanned hosts which have IPv4 addresses that are mapped to IPv6 addresses in your account, you enter the IPv4 addresses for the ips parameter.

For example, if the special IPv4 address 0.0.0.199 is mapped to an IPv6 address in your account and this IP address has been scanned, you can make this API request:

```bash
curl -H "X-Requested-With: Curl Sample" -u "username:password"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?
action=list&ips=0.0.0.100"
```

XML output returned will show the IPv4 address and the IPv6 address for the host, as shown below (XML fragment):

```xml
...<HOST>
  <ID>276010</ID>
  <IP>0.0.0.100</IP>
  <IPV6>2001:470:8418:a18::a0a:18c7</IPV6>
  <TRACKING_METHOD>IP</TRACKING_METHOD>
  <OS><![CDATA[Windows 2003 Service Pack 2]]></OS>
  <DNS><![CDATA[mssql2k8-24-199.patch.qualys.com]]></DNS>
  <LAST_SCAN_DATETIME>2018-06-17T19:06:31Z</LAST_SCAN_DATETIME>
  <DETECTION_LIST>
  ...
</HOST>
...```

### IPv6 Mapping Record List

/`api/2.0/fo/asset/ip/v4_v6`

[GET] [POST]

View a list of IPv6 mapping records in the subscription. Each mapping record associates one IPv6 address in your network with one IPv4 address in the special mapping range 0.0.0.1-0.254.255.255.

A maximum of 5,000 IPv6 mapping records will be processed per request, unless the truncation_limit input parameter is specified. If the requested list identifies more than 5,000 records or the number of records specified using truncation_limit, then the XML output includes the `<WARNING>` element and instructions for making another request for the next batch of records.

Permissions - Managers can view all IPv6 mapping records when the IPv6 Support feature is enabled for the user’s subscription. Other users do not have permission to view IPv6 mapping records.
Add IPv6 Mapping Records

/api/2.0/fo/asset/ip/v4_v6

[POST]

Add IPv6 mapping records to the subscription. Each mapping record associates one IPv6 address in your network with one IPv4 address in the special mapping range 0.0.0.1-0.254.255.255. A maximum of 10,000 mapping records can be added per API request.

### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(Optional) Show only mapping records which have a minimum record ID. A valid mapping record ID is required. When unspecified, records are not filtered by record ID.</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(Optional) Show only mapping records which have a maximum record ID. A valid mapping record ID is required.</td>
</tr>
<tr>
<td>ipv4_filter={value}</td>
<td>(Optional) Show only mapping records with certain IPv4 addresses. When unspecified, records are not filtered by IPv4 addresses.</td>
</tr>
<tr>
<td>ipv6_network={value}</td>
<td>(Optional) Show only mapping records with certain IPv6 network addresses. When unspecified, records are not filtered by IPv6 network addresses.</td>
</tr>
<tr>
<td>output_format={CSV</td>
<td>XML}</td>
</tr>
<tr>
<td>truncation_limit={value}</td>
<td>(Optional) The maximum number of mapping records to be returned by the API request. A valid value is an integer between 1 and 1,000,000. When unspecified, 5,000 records will be returned.</td>
</tr>
</tbody>
</table>

### DTD

<platform API server>/api/2.0/fo/asset/ip/v4_v6/asset/ip/v4_v6/ip_map_list_output.dtd

### Sample IPv6 Mapping Records List Output

How to Add IPv6 Records in CSV
How to Add IPv6 Records in XML
Permissions - Managers can add IPv6 mapping records, when the IPv6 Support feature is enabled for the user’s subscription. Other user roles do not have these permissions.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=add</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0</td>
<td>1]</td>
</tr>
<tr>
<td>csv_data={value}</td>
<td>The CSV data file containing the IPv6 mapping records that you want to add. This parameter or xml_data must be specified. See How to Add IPv6 Records in CSV. The parameters csv_data and xml_data cannot be specified in the same request.</td>
</tr>
<tr>
<td>xml_data={value}</td>
<td>The CSV data file containing the IPv6 mapping records that you want to add. This parameter or csv_data must be specified. See How to Add IPv6 Records in XML. The parameters csv_data and xml_data cannot be specified in the same request.</td>
</tr>
<tr>
<td>all_or_nothing=0</td>
<td>1]</td>
</tr>
</tbody>
</table>

**DTD**

`<platform API server>/api/2.0/simple_return.dtd`

**Sample XML Output**

How to Add IPv6 Records in CSV

How to Add IPv6 Records in XML
Networks

The Network API is used to manage networks when the Network Support feature is enabled in the user’s subscription.

Network List
Create Network
Update Network
Assign Scanner Appliance to Network

Network List

/api/2.0/fo/network/?action=list

[GET] [POST]

List custom networks in your account.

Permissions - A Manager will view all custom networks in the subscription, a Unit Manager will view custom networks in their business unit’s assigned asset groups, and a Scanner/Reader will view custom networks in their account’s assigned asset groups.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>ids={value1,value2}</td>
<td>(Optional) Filter the list to view specific networks.</td>
</tr>
</tbody>
</table>

Sample - List custom networks

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/network/?action=list&ids=7343,7345,7350"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE NETWORK_LIST SYSTEM "https://qualysapi.qualys.com/network_list_output.dtd">
RESPONSE
```
<DATETIME>2018-05-28T01:06:45Z</DATETIME>

NETWORK_LIST

NETWORK

-ID>7343</ID>

<NAME>&lt;![CDATA[My New Network]]&gt;</TITLE>

SCANNER_APPLIANCE_LIST

SCANNER_APPLIANCE

-ID>1234</ID>

<FRIENDLY_NAME>&lt;![CDATA[abc123]]&gt;</FRIENDLY_NAME>
</SCANNER_APPLIANCE>
</SCANNER_APPLIANCE_LIST>
</NETWORK>
...

</NETWORK_LIST>
</RESPONSE>

DTD

<platform API server>/api/2.0/fo/network/network_list_output.dtd

Create Network

/api/2.0/fo/network/?action=create

[POST]

Create a new custom network.

Permissions - This API is available to Managers only.

Know more - Before you’re ready to start scanning, you’ll need to 1) assign scanner appliance(s) to your network, and 2) add host assets to your network (assign asset groups to it).

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=create</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>name=[value]</td>
<td>(Required) A user-defined friendly name for your network. A successful request will return a unique network ID and this is used to manage your network using the API.</td>
</tr>
</tbody>
</table>
Sample - Create custom network

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=create&name=My+Network" "https://qualysapi.qualys.com/api/2.0/fo/network/"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2018-01-14T04:37:24Z</DATETIME>
    <TEXT>Network created with ID</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>id</KEY>
        <VALUE>1103</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

DTD
```
<platform API server>/api/2.0/simple_return.dtd
```
Update Network

/api/2.0/fo/network/?action=update

[POST]

Create a new custom network.

Permissions - This API is available to Managers only.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0</td>
<td>(Optional) Show (echo) the request’s input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.</td>
</tr>
<tr>
<td>name={value}</td>
<td>(Required) Specify a new network name. (The network ID is assigned by our service and it can’t be changed.)</td>
</tr>
</tbody>
</table>

Sample - Update network

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "id=1130&action=update&name=Network+123" "https://qualysapi.qualys.com/api/2.0/fo/network/"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2018-05-20T06:17:06Z</DATETIME>
        <TEXT>Network updated</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>id</KEY>
                <VALUE>1103</VALUE>
            </ITEM>
            <ITEM>
                <KEY>name</KEY>
                <VALUE>Network 123</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```
Assign Scanner Appliance to Network

/api/2.0/fo/appliance/?action=assign_network_id

[POST]

Assign a scanner appliance to a network. When the network support feature is enabled for your subscription, scanner appliances are assigned to networks. Each appliance can be assigned to 1 network only.

Permissions - This API is available to Managers only.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=assign_network_id</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>appliance_id={value}</td>
<td>(Required) ID of the scanner appliance you want to assign to a network.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Required) ID of the network you want to assign the scanner appliance to.</td>
</tr>
</tbody>
</table>

Sample - Assign scanner appliance to network

API request:


XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
 <RESPONSE>
```

594
<DATETIME>2018-03-16T22:50:49Z</DATETIME>
<TEXT>Success: Network ID=[1103] assigned to Appliance with ID=[506]</TEXT>
</RESPONSE>
</SIMPLE_RETURN>

Or, if unsuccessful, the response might look like this:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
<RESPONSE>
<DATETIME>2018-03-16T22:53:41Z</DATETIME>
<CODE>1905</CODE>
<TEXT>parameter network_id has invalid value: 1103 (No such network ID)</TEXT>
</RESPONSE>
</SIMPLE_RETURN>

DTD
<platform API server>/api/2.0/simple_return.dtd
Reports

Launch and manage reports in your account. Report Share must be enabled for your account.

Report List
Launch Report
Launching Reports Using Asset Tags
Launching and Fetching Compliance Reports in CSV Format
Report Template List
Launch Scorecard
Cancel Running Report
Download Saved Report
Delete Saved Report
Scheduled Reports List
Launch Scheduled Report
Asset Search Report
**Report List**

/api/2.0/fo/report/?action=list

[GET] [POST]

View a list of reports in the user's account when Report Share feature is enabled. The report list output includes all report types, including scorecard reports.

User permissions - Managers and Auditors view all assets in the subscription, Unit Managers view assets in their own business unit, Scanners and Readers view assets in their own account.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Optional) Specifies a report ID of a report that is saved in the Report Share storage space. When specified, information on the selected report will be included in the XML output.</td>
</tr>
<tr>
<td>state={value}</td>
<td>(Optional) Specifies that reports with a certain state will be included in the XML output. By default, all states are included. A valid value is: Running (reports are in progress), Finished, Submitted, Canceled, or Errors.</td>
</tr>
<tr>
<td>user_login={value}</td>
<td>(Optional) Specifies a user login ID. This parameter is used to restrict the XML output to reports launched by the specified user login ID.</td>
</tr>
<tr>
<td>expires_before_datetime={date}</td>
<td>(Optional) Specifies the date and time (optional) when reports will expire in the future. Only reports that expire before this date/time will be included in the XML output. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”.</td>
</tr>
<tr>
<td>client_id={value}</td>
<td>(Optional) Id assigned to the client (Consultant type subscriptions).</td>
</tr>
<tr>
<td>client_name={value}</td>
<td>(Optional) Name of the client (Consultant type subscriptions). Note: The client_id and client_name parameters are mutually exclusive and cannot be specified together in the same request.</td>
</tr>
</tbody>
</table>
Sample - List reports

curl -H "X-Requested-With: Curl Sample"
-k "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/report/
?action=list"

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE REPORT_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/report/report_list_output .dtd">

<REPORT_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-10-30T22:32:15Z</DATETIME>
    <REPORT_LIST>
      <REPORT>
        <ID>42703</ID>
        <TITLE><![CDATA[Test now]]></TITLE>
        <TYPE>Scan</TYPE>
        <USER_LOGIN>acme_aa</USER_LOGIN>
        <LAUNCH_DATETIME>2017-10-30T17:59:22Z</LAUNCH_DATETIME>
        <OUTPUT_FORMAT>PDF</OUTPUT_FORMAT>
        <SIZE>129.1 MB</SIZE>
        <STATUS>
          <STATE>Finished</STATE>
        </STATUS>
        <EXPIRATION_DATETIME>2017-11-06T17:59:24Z</EXPIRATION_DATETIME>
      </REPORT>
      <REPORT>
        <ID>42700</ID>
        <TYPE>Scorecard</TYPE>
        <USER_LOGIN>acme_ts2</USER_LOGIN>
        <LAUNCH_DATETIME>2017-10-29T22:12:42Z</LAUNCH_DATETIME>
        <OUTPUT_FORMAT>SECURE_PDF</OUTPUT_FORMAT>
        <SIZE>18.1 KB</SIZE>
        <STATUS>
          <STATE>Finished</STATE>
        </STATUS>
        <EXPIRATION_DATETIME>2017-11-05T17:59:24Z</EXPIRATION_DATETIME>
      </REPORT>
      <REPORT>
        <ID>42699</ID>
        <TYPE>Scorecard</TYPE>
        <USER_LOGIN>quays_ts2</USER_LOGIN>
        ...
Launch a report in the user's account. The Report Share feature must be enabled in the user's subscription. When a report is launched with Report Share, the report is run in the background, and the report generation processing does not timeout until the report has completed.

User permissions - Managers and Auditors can launch scorecard reports on all assets in the subscription, Unit Managers can launch scorecard reports on assets in their own business unit, Scanners and Readers can launch scorecard reports on assets in their own account.

Note: The Launch Report API for Compliance Policy Reports is available as part of one of the following subscription combinations only:

- PC and API add-on
- PC, SCA, and API add-on
- VMDR, SCA, and API add-on
### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>action=launch</code></td>
<td>(Required)</td>
</tr>
<tr>
<td>`echo_request=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>template_id={value}</code></td>
<td>(Required) The template ID of the report you want to launch. Use the /msp/report_template_list.php API to find the template ID you’re interested in. See Report Template List.</td>
</tr>
<tr>
<td><code>report_title={value}</code></td>
<td>(Optional) A user-defined report title. The title may have a maximum of 128 characters. For a PCI compliance report, the report title is provided by Qualys and cannot be changed.</td>
</tr>
<tr>
<td><code>output_format={value}</code></td>
<td>(Required) One output format may be specified. Supported formats for various reports are below. - map report: pdf, html (a zip file), mht, xml, or csv - scan report: pdf, html (a zip file), mht, xml, csv, or docx - remediation report: pdf, html (a zip file), mht, or csv - compliance report (not PCI): pdf, html (a zip file), or mht - PCI compliance report: pdf or html (a zip file) - patch report: pdf, online, xml or csv - compliance policy report: pdf, html (a zip file), mht, xml, or csv (see Launching and Fetching Compliance Reports in CSV Format)</td>
</tr>
<tr>
<td>`hide_header=[0</td>
<td>1]`</td>
</tr>
<tr>
<td><code>pdf_password={value}</code></td>
<td>(Required for secure PDF distribution, Manager or Unit Manager only) The password to be used for encryption. Requirements: - the password must have a minimum of 8 characters (ascii), and a maximum of 32 characters - the password must contain alpha and numeric characters - the password cannot match the password for the user’s Qualys account. - the password must follow the password security guidelines defined for your subscription (log into your account and go to Users &gt; Setup &gt; Security)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>recipient_group={value}</td>
<td>(Optional for secure PDF distribution, Manager or Unit Manager only) The report recipients in the form of one or more distribution group names, as defined using the Qualys UI. Multiple distribution groups are comma separated. A maximum of 50 distribution groups may be entered. The recipient_group parameter can only be specified when the pdf_password parameter is also specified. The recipient_group parameter cannot be specified in the same request as recipient_group_id.</td>
</tr>
<tr>
<td>recipient_group_id={value}</td>
<td>(Optional for secure PDF distribution, Manager or Unit Manager only) The report recipients in the form of one or more distribution group IDs. Multiple distribution group IDs are comma separated. Where do I find this ID? Log in to your Qualys account, go to Users &gt; Distribution Groups and select Info for a group in the list. The recipient_group_id parameter can only be specified when the pdf_password parameter is also specified. The recipient_group_id parameter cannot be specified in the same request as recipient_group.</td>
</tr>
</tbody>
</table>

**MAP REPORT**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_type=Map</td>
<td>(Optional)</td>
</tr>
<tr>
<td>domain={value}</td>
<td>(Required for map report) Specifies the target domain for the map report. Include the domain name only; do not enter “www.” at the start of the domain name. When the special “none” domain is specified as a parameter value, the ip_restriction parameter is required.</td>
</tr>
<tr>
<td>ip_restriction={value}</td>
<td>(Optional for map report) For a map report, specifies certain IPs/ranges to include in the report. This parameter is required when the domain parameter is specified with the value “none” (for the special &quot;none&quot; domain). Multiple IPs and/or ranges are comma separated.</td>
</tr>
<tr>
<td>report_refs={value}</td>
<td>(Required for map report) For a map report, specifies the map references (1 or 2) to include. A map reference starts with the string “map/” followed by a reference ID number. When two map references are given, the report compares map results. Two map references are comma separated.</td>
</tr>
</tbody>
</table>

**SCAN REPORT - SCAN BASED FINDINGS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_type=Scan</td>
<td>(Optional)</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_refs={value}</td>
<td>(Required for Manual scan report) For a Manual scan report, this parameter specifies the scan references to include. A scan reference starts with the string “scan/” followed by a reference ID number. Multiple scan references are comma separated.</td>
</tr>
<tr>
<td>ip_restriction={value}</td>
<td>(Optional for Manual scan report) For a scan report, the report content will be restricted to the specified IPs/ranges. Multiple IPs and/or ranges are comma separated.</td>
</tr>
</tbody>
</table>

**SCAN REPORT - HOST BASED FINDINGS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_type=Scan</td>
<td>(Optional)</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional) Specify IPs/ranges to change (overwrite) the report target, as defined in the report template. Multiple IPs/ranges are comma separated. When specified, hosts defined in the report template are not included in the report.</td>
</tr>
<tr>
<td></td>
<td>You can specify ips and/or asset_group_ids, or asset tags (see Launching Reports Using Asset Tags).</td>
</tr>
<tr>
<td>asset_group_ids={value}</td>
<td>(Optional) Specify asset group IDs to change (overwrite) the report target, as defined in the report template. When specified, hosts defined in the report template are not included in the report.</td>
</tr>
<tr>
<td></td>
<td>You can specify ips and/or asset_group_ids, or asset tags (see Launching Reports Using Asset Tags).</td>
</tr>
<tr>
<td>ips_network_id={value}</td>
<td>(Optional, and valid only when the Network Support feature is enabled for the user’s account) The ID of a network that is used to restrict the report’s target to the IPs/ranges specified in the “ips” parameter. Set to a custom network ID (note this does not filter IPs/ranges specified in “asset_group_ids”). Or set to “0” (the default) for the Global Default Network - this is used to report on hosts outside of your custom networks.</td>
</tr>
</tbody>
</table>

**PATCH REPORT**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ips={value}</td>
<td>(Optional for patch report) Specify IPs/ranges to change (override) the report target, as defined in the patch report template. Multiple IPs/ranges are comma separated. When specified, hosts defined in the report template are not included in the report.</td>
</tr>
<tr>
<td></td>
<td>You can specify ips and/or asset_group_ids, or asset tags (see Launching Reports Using Asset Tags).</td>
</tr>
</tbody>
</table>
### Reports

#### Launch Report

**asset_group_ids={value}** (Optional for patch report) Specify IPs/ranges to change (override) the report target, as defined in the patch report template. Multiple asset group IDs are comma separated. When specified, hosts defined in the report template are not included in the report.

You can specify ips and/or asset_group_ids, or asset tags (see Launching Reports Using Asset Tags).

#### REMEDIATION REPORT

**report_type=Remediation** (Optional)

**ips={value}** (Optional for remediation report) Specify IPs/ranges you want to include in the report. Multiple IPs and/or ranges are comma separated.

You can specify ips and/or asset_group_ids, or asset tags (see Launching Reports Using Asset Tags).

**asset_group_ids={value}** (Optional for remediation report) Specify asset group IDs that identify hosts you want to include in the report. Multiple asset group IDs are comma separated.

You can specify ips and/or asset_group_ids, or asset tags (see Launching Reports Using Asset Tags).

**assignee_type={User|All}** (Optional for remediation report) Specifies whether the report will include tickets assigned to the current user (User is set by default), or all tickets in the user account. By default tickets assigned to the current user are included.

#### COMPLIANCE REPORT

**report_type=Compliance** (Optional) For compliance type report. Compliance type reports are Qualys Top 20 Report, SANS Top 20 Report, Qualys PCI Executive Report, and Qualys PCI Technical Report.

**ips={value}** (Optional for compliance report) For a compliance report (except a PCI report), specify the IPs/ranges you want to include in the report. Multiple IPs and/or ranges are comma separated.

You can specify ips and/or asset_group_ids, or asset tags (see Launching Reports Using Asset Tags).

Optional: Qualys Top 20 Report, SANS Top 20 Report

Invalid: PCI Executive Report, PCI Technical Report
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asset_group_ids={value}</td>
<td>(Optional for compliance report) For a compliance report (except a PCI report), specify asset groups IDs which identify hosts to include in the report. Multiple asset group IDs are comma separated.</td>
</tr>
<tr>
<td></td>
<td>You can specify ips and/or asset_group_ids, or asset tags (see Launching Reports Using Asset Tags).</td>
</tr>
<tr>
<td></td>
<td>Optional: Qualys Top 20 Report, SANS Top 20 Report</td>
</tr>
<tr>
<td></td>
<td>Invalid: PCI Executive Report, PCI Technical Report</td>
</tr>
<tr>
<td>report_refs={value}</td>
<td>(Required for PCI compliance report) For a PCI compliance report, either the technical or executive report, this parameter specifies the scan reference to include. A scan reference starts with the string “scan/” followed by a reference ID number. The scan reference must be for a scan that was run using the PCI Options profile. Only one scan reference may be specified.</td>
</tr>
<tr>
<td></td>
<td>Required: PCI Executive Report, PCI Technical Report</td>
</tr>
<tr>
<td></td>
<td>Invalid: Qualys Top 20 Report, SANS Top 20 Report</td>
</tr>
</tbody>
</table>

**COMPLIANCE POLICY REPORT**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report_type=Policy</td>
<td>(Optional)</td>
</tr>
<tr>
<td>policy_id={value}</td>
<td>(Required) Specifies the policy to run the report on. A valid policy ID must be entered.</td>
</tr>
<tr>
<td>asset_group_ids={value}</td>
<td>(Optional) Specify asset group IDs if you want to include only certain asset groups in your report. These asset groups must be assigned to the policy you are reporting on. Multiple asset group IDs are comma separated.</td>
</tr>
<tr>
<td></td>
<td>You can specify ips and/or asset_group_ids, or asset tags (see Launching Reports Using Asset Tags).</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional) Specify IPs/ranges if you want to include only certain IP addresses in your report. These IPs must be assigned to the policy you’re reporting on. Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>You can specify ips and/or asset_group_ids, or asset tags (see Launching Reports Using Asset Tags).</td>
</tr>
<tr>
<td>host_id={value}</td>
<td>(Optional) In the policy report output, show only results for a single host instance. Specify the ID for the host to include in the report. A valid host ID must be entered.</td>
</tr>
<tr>
<td></td>
<td>This parameter must be specified with instance_string.</td>
</tr>
</tbody>
</table>
**Launch Report**

**Sample - Launch Report**

```
curl -H "X-Requested-With: Curl Sample"
   -d "action=launch&template_id=55469&output_format=pdf"
   -b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
   secure" "https://qualysapi.qualys.com/api/2.0/fo/report/"
```

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE GENERIC SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
   <RESPONSE>
      <DATETIME>2017-06-20T21:45:23Z</DATETIME>
      <TEXT>New report launched</TEXT>
      <ITEM_LIST>
         <ITEM>
            <KEY>ID</KEY>
            <VALUE>1665</VALUE>
         </ITEM>
      </ITEM_LIST>
   </RESPONSE>
</SIMPLE_RETURN>
```

**Parameter** | **Description**
--- | ---
instance_string={value} | (Optional) Specifies a single instance on the selected host. The instance string may be "os" or a string like "oracle10:1:1521:ora10204u". Use the Compliance Posture Information API (with the endpoint/api/2.0/fo/compliance/posture/info) to find the appropriate instance string. This parameter must be specified with host_id.

**DTD**

<platform API server>/api/2.0/simple_return.dtd
Launching Reports Using Asset Tags

It’s possible to select asset tags for both vulnerability and compliance reports. Use the following tag parameters to launch your report using asset tags.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>use_tags={0</td>
<td>1}</td>
</tr>
<tr>
<td>tag_include_selector={all</td>
<td>any}</td>
</tr>
<tr>
<td>tag_exclude_selector={all</td>
<td>any}</td>
</tr>
<tr>
<td>tag_set_by={id</td>
<td>name}</td>
</tr>
<tr>
<td>tag_set_include={value}</td>
<td>(Optional) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. tag_set_include is valid only when use_tags=1 is specified.</td>
</tr>
<tr>
<td>tag_set_exclude={value}</td>
<td>(Optional) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. tag_set_exclude is valid only when use_tags=1 is specified.</td>
</tr>
</tbody>
</table>

**API request:**

```bash
```

**XML output:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE GENERIC SYSTEM ""
```
Launching and Fetching Compliance Reports in CSV Format

Policy Compliance Reporting Service (PCRS) is a new reporting service to improve performance in Policy Compliance report generation. With PCRS, we’ve enhanced policy reports in CSV format by automatically compressing large size reports. When you run a policy report in CSV format, the report will be in ZIP format if the report size is between 1 GB and 5 GB; while reports less than 1 GB will be in CSV format. Similar improvements to other report formats will be added soon. You can download reports from the user interface or fetch reports by using APIs.

**Note:** This feature will be automatically enabled for customers with the release of Qualys Policy Compliance Reporting Service 1.0.0. Contact Qualys Support if you do not want this feature to be enabled for your subscription.

**Important:** If you are currently using the Report API to launch and fetch compliance policy reports in CSV format, then it’s important to note that once PCRS is enabled for your subscription, any CSV compliance policy report that is over 1GB in size will be compressed automatically and you will get a ZIP file instead of a CSV file. You’ll need to update your code or work with your 3rd party vendor to monitor the response header and if the report is compressed, add a step to uncompress the ZIP file before parsing the data.

When fetching a report using the API, the response header will indicate if the report is compressed or not. See the API samples that follow.

- In case of compressed reports, header content-type is - application/zip
- In case of uncompressed reports, header content-type is - text/csv

**Sample: Report size more than 1 GB**

In this sample, the report being downloaded is more than 1GB in size.
API Request

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -d "action=fetch&id=<REPORT ID>"
"https://qualysapi.qualys.com/api/2.0/fo/report/"
```

Response Header

You'll notice that the header Content-Type is "application/zip"

* About to connect() to qualysapi.xxx.qualys.com port <PORT NUMBER> (#0)
  * Trying xx.xx.x.xxx ...
  * Connected to qualysapi.xxx.qualys.com (xx.xx.x.xxx) port <PORT NUMBER> (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate:
  * subject: CN=*.xxx.qualys.com,OU=Engineering,O="Qualys, Inc.",L=Foster City,ST=California,C=US
  * start date: Sep 16 09:45:00 2020 GMT
  * expire date: Sep 16 09:45:00 2022 GMT
  * common name: *.xxx.qualys.com
  * issuer: E=xx@qualys.com,CN=Qualys Ops T2v1,OU=Operations,O="Qualys, Inc.",L=Redwood City,ST=California,C=US
* Server auth using Basic with user '<USER NAME>'
> POST /api/2.0/fo/report/ HTTP/1.1
> Authorization: <AUTHORIZATION TOKEN>
> User-Agent: curl/7.29.0
> Host: qualysapi.xxx.qualys.com
> Accept: */*
> X-Requested-With:curl demo2
> Content-Length: 22
> Content-Type: application/x-www-form-urlencoded
> }
[data not shown]
* upload completely sent off: 22 out of 22 bytes
< HTTP/1.1 200 OK
< Date: Thu, 07 Oct 2021 11:15:03 GMT
< Server: Qualys
< Strict-Transport-Security: max-age=63072000;
< X-XSS-Protection: 1; mode=block
< X-Content-Type-Options: nosniff
< X-Frame-Options: SAMEORIGIN
< Strict-Transport-Security: max-age=31536000; includeSubDomains
< X-RateLimit-Limit: 300
< X-RateLimit-Window-Sec: 3600
< X-Concurrency-Limit-Limit: 2
< X-Concurrency-Limit-Running: 0
< X-RateLimit-ToWait-Sec: 0
< X-RateLimit-Remaining: 297
< Content-Length: 221540169
< Connection: keep-alive
< Content-Disposition: attachment; filename=<FILENAME>.zip
< Content-Type: application/zip
Sample: Report size less than 1 GB
In this sample, the report being downloaded is less than 1GB in size.

API Request


Response Header

You’ll notice that the header Content-Type is "text/csv;charset=UTF-8"

* About to connect() to qualysapi.xxx.qualys.com port <PORT NUMBER> (#0)
* Trying xx.xx.x.xxxx...
* Connected to qualysapi.xxx.qualys.com (xx.xx.x.xxxx) port <PORT NUMBER> (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate:  
  * subject: CN=*.*.xxx.qualys.com,OU=Engineering,O="Qualys, Inc.",L=Foster City,ST=California,C=US
  * start date: Sep 16 09:45:00 2020 GMT
  * expire date: Sep 16 09:45:00 2022 GMT
  * common name: *.*.xxx.qualys.com
  * issuer: E=xx@qualys.com,CN=Qualys Ops T2v1,OU=Operations,O="Qualys, Inc.",L=Redwood City,ST=California,C=US
* Server auth using Basic with user 'user name'
> POST /api/2.0/fo/report/ HTTP/1.1
> Authorization: <AUTHORIZATION TOKEN>
> User-Agent: curl/7.29.0
> Host: qualysapi.xxx.qualys.com
> Accept: */*
> X-Requested-With:curl demo2
> Content-Length: 22
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 22 out of 22 bytes
< HTTP/1.1 200 OK
< Date: Thu, 07 Oct 2021 11:16:21 GMT
< Server: Qualys
< Strict-Transport-Security: max-age=63072000;
< X-XSS-Protection: 1; mode=block
< X-Content-Type-Options: nosniff
< X-Frame-Options: SAMEORIGIN
< Strict-Transport-Security: max-age=31536000; includeSubDomains
< X-Rate-Limit-Limit: 300
< X-Rate-Limit-Window-Sec: 3600
< X-Concurrency-Limit-Limit: 2
< X-Concurrency-Limit-Running: 0
< X-Rate-Limit-ToWait-Sec: 0
< X-Rate-Limit-Remaining: 296
< Content-Length: 294850
Report Template List

/msp/report_template_list.php

[GET] [POST]

List available report templates, including template titles and IDs, in the user account. The report list includes templates for all report types.

DTD
<platform API server>/report_template_list.dtd

Sample - Report template list

API request:

curl -u username:password -H "X-Requested-With: curl"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE REPORT_TEMPLATE_LIST SYSTEM
"https://qualysapi.qualys.com/report_template_list.dtd">
<REPORT_TEMPLATE_LIST>
  <REPORT_TEMPLATE>
    <ID>235288</ID>
    <TYPE>Auto</TYPE>
    <TEMPLATE_TYPE>Scan</TEMPLATE_TYPE>
    <TITLE><![CDATA[Windows Authentication QIDs]]></TITLE>
    <USER>
      <LOGIN><![CDATA[acme_jk]]></LOGIN>
      <FIRSTNAME><![CDATA[Jason]]></FIRSTNAME>
      <LASTNAME><![CDATA[Kim]]></LASTNAME>
    </USER>
    <LAST_UPDATE>2018-02-12T18:09:10Z</LAST_UPDATE>
    <GLOBAL>0</GLOBAL>
  </REPORT_TEMPLATE>
  <REPORT_TEMPLATE>
    <ID>235164</ID>
    <TYPE>Auto</TYPE>
    <TEMPLATE_TYPE>Policy</TEMPLATE_TYPE>
    ...
Each <REPORT_TEMPLATE> element identifies template properties, including the report template ID, template type and title, in the sub-elements described below.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ID&gt;</td>
<td>The template ID number.</td>
</tr>
<tr>
<td>&lt;TYPE&gt;</td>
<td>The template type: Auto (for automatic) or Manual.</td>
</tr>
</tbody>
</table>
Launch Scorecard

/\api/2.0/fo/report/scorecard

[POST]

Launch a vulnerability scorecard report in the user’s Report Share. It is not possible to launch any compliance scorecard reports or WAS scorecard reports using this API at this time.

When a scorecard report is launched, the report is run in the background, and the report generation processing does not timeout until the report has completed.

User Permissions - Managers and Auditors can launch scorecard reports on all assets in the subscription, Unit Managers can launch scorecard reports on assets in their own business unit, Scanners and Readers can launch scorecard reports on assets in their own account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=launch</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0}[1]</td>
<td>(Optional) Specifies whether to echo the request’s input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>name=[value]</td>
<td>(Required) Specifies the scorecard name for the vulnerability scorecard report that you want to launch. This name corresponds to a service-provided scorecard or a user-created scorecard. For a service-provided scorecard, specify one of these names: Asset Group Vulnerability Report Ignored Vulnerabilities Report Most Prevalent Vulnerabilities Report Most Vulnerable Hosts Report Patch Report</td>
</tr>
<tr>
<td>report_title=[value]</td>
<td>(Optional) Specifies a user-defined report title. The title may have a maximum of 128 characters. When unspecified, the report title will be the scorecard name.</td>
</tr>
</tbody>
</table>
| output_format=[value] | (Required) Specifies the output format of the report. One output format may be specified. A valid value is: pdf, html (a zip file), mht, xml, or csv.  
When output_format=pdf is specified, the Secure PDF Distribution may be used. See “Sample - Launch Report.” |
| hide_header=[0][1] | (Valid for CSV format report only). Specify hide_header=1 to omit the header information from the report. By default this information is included. |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| pdf_password={value}       | (Required for secure PDF distribution, Manager or Unit Manager only) The password to be used for encryption. The password may have a maximum of 32 characters (ascii). The password cannot match the password for the user’s Qualys login account. The password must follow the password security guidelines defined for the user’s subscription.  
  Conditions:  
  a) The pdf_password parameter can only be specified by a Manager or Unit Manager.  
  b) The pdf_password parameter can only be specified when Report Share is enabled for your subscription and the option “Enable Secure PDF Distribution” is selected (log into your account and go to Users > Setup > Security). |
| recipient_group={value}    | (Optional for secure PDF distribution, Manager or Unit Manager only)  
  The report recipients in the form of one or more distribution group names, as defined in your Qualys account. Each distribution group identifies a list of users who will receive the secure PDF report. Multiple distribution groups are comma separated. A maximum of 50 distribution groups may be entered.  
  Conditions:  
  a) The recipient_group parameter can only be specified when the pdf_password parameter is also specified.  
  b) The recipient_group parameter can only be specified by a Manager or Unit Manager.  
  c) The recipient_group parameter can only be specified when Report Share is enabled for your subscription and the option “Enable Secure PDF Distribution” is selected (Setup—>Report Share).  
  d) The recipient_group parameter cannot be specified in the same request as recipient_group_id |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| recipient_group_id=[value] | (Optional for secure PDF distribution, Manager or Unit Manager only) The report recipients in the form of one or more distribution group IDs. Multiple distribution group IDs are comma separated. Where do I find this ID? Log in to your Qualys account, go to Users > Distribution Groups and select Info for a group in the list. Conditions:  
  a) The recipient_group_id parameter can only be specified when the pdf_password parameter is also specified.  
  b) The recipient_group_id parameter can only be specified by a Manager or Unit Manager.  
  c) The recipient_group_id parameter can only be specified when Report Share is enabled for your subscription and the option "Enable Secure PDF Distribution" is selected (Setup—>Report Share).  
  d) The recipient_group_id parameter cannot be specified in the same request as recipient_group. |
| source=[value]         | (Conditional) The source asset groups for the report. Specify asset_groups to select asset groups. Specify business_unit to select all the asset groups in a business unit.  
  For a user scorecard, this parameter is optional. When unspecified, the source selection set in the scorecard attributes (as defined in your Qualys account) is used. Conditions:  
  a) The source parameter is required for a service-provided scorecard.  
  b) For a user scorecard, the source selection specified in the source parameter replaces an existing source selection set in the scorecard attributes (as defined in your Qualys account). If you set this parameter to asset_groups, you must specify one of these parameters: asset_groups or all_asset_groups. If you set this parameter to business_unit then you must specify one or more of these parameters: business_unit, division, function and/or location. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| asset_groups={value}   | (Conditional) The titles of asset groups to be used as source asset groups for the scorecard report. One or more asset group titles in your account may be specified. Multiple asset group titles are comma separated. Conditions:  
  a) The asset_groups parameter can only be specified when source=asset_groups.  
  b) These parameters cannot be specified for the same API request: asset_groups and all_asset_groups. |
| all_asset_groups={1}   | (Conditional) Set to 1 to select all asset groups available in your account as the source asset groups for the scorecard report. Conditions:  
  a) The asset_groups parameter can only be specified when source=asset_groups.  
  b) These parameters cannot be specified for the same API request: asset_groups and all_asset_groups. |
| business_unit={value}  | (Conditional for a Manager; not valid for other users) The title of a business unit containing the source asset groups for the scorecard report. All asset groups in the business unit will be included in the report source. You may enter the title of a business unit in your account that was created by a Manager user, or you may enter “Unassigned” for the unassigned business unit. For a user scorecard, the business unit replaces an existing business unit set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (business_unit=), the existing business unit in the scorecard attributes is not included in the scorecard parameters submitted with the API request. Conditions:  
  a) When source=business_unit, one or more of these parameters must be specified: business_unit, division, function and/or location.  
  b) The business_unit parameter can only be specified by a Manager. |
## Parameter Description

**division={value}**  
(Conditional) A business info tag identifying a division that asset group(s) belong to. The tag must be defined for an asset group in your account. When specified, only asset groups with this tag are included in the scorecard report source.

For a user scorecard, the division tag replaces an existing tag set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (division=), the existing division tag in the scorecard attributes is not included in the scorecard parameters submitted with the API request.

**Conditions:**  
a) When source=business_unit, one or more of these parameters must be specified: business_unit, division, function and/or location.

b) The division parameter can only be specified when source=business_unit.

**function={value}**  
(Conditional) A business info tag identifying a business function for asset group(s). The tag must be defined for an asset group in your account. When specified, only asset groups with this tag are included in the scorecard report source.

For a user scorecard, the function tag replaces an existing function tag set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (function=), the existing function tag in the scorecard attributes is not included in the scorecard parameters submitted with the API request.

**Conditions:**  
a) When source=business_unit, one or more of these parameters must be specified: business_unit, division, function and/or location.

b) The function parameter can only be specified when source=business_unit.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>location={value}</td>
<td>(Conditional) A business info tag identifying a location where asset group(s) are located. The tag must be defined for an asset group in your account. When specified, only asset groups with this tag are included in the scorecard report source. For a user scorecard, the location tag replaces an existing location tag set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (location=), the existing location tag in the scorecard attributes is not included in the scorecard parameters submitted with the API request. Conditions: a) When source=business_unit, one or more of these parameters must be specified: business_unit, division, function and/or location. b) The location parameter can only be specified when source=business_unit.</td>
</tr>
<tr>
<td>patch_qids={value}</td>
<td>(Conditional for Patch Report scorecard; not valid for other scorecards) Up to 10 QIDs for vulnerabilities or potential vulnerabilities with available patches. Multiple QIDs are comma separated. When the QIDs are detected on a host this means the host does not have the patches installed and it will be reported in the scorecard output. For a user-defined Patch Report, the patch QIDs list replaces the patch QIDs list set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (patch_qids=), the existing patches QIDs list in the scorecard attributes is not included in the scorecard parameters submitted with the API request. Conditions: a) The patch_qids parameter may be specified only for a Patch Report. b) For a Patch Report, patch_qids or missing_qids must be specified. Both parameters may be specified together.</td>
</tr>
</tbody>
</table>
Cancel Running Report

/\api/2.0/fo/report

[POST]

Cancel a running report in the user’s account. This is an option when Report Share is enabled in the user’s subscription.

User permissions - Managers can cancel any running report. Unit Managers can cancel a running report in their own business unit (report launched by user in their own business unit). Scanners and Readers can cancel their own running report.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>missing_qids={value}</td>
<td>(Conditional for Patch Report scorecard; not valid for other scorecards) One or two QIDs for missing software. Two QIDs are comma separated. Typically missing software QIDs are information gathered checks. When the QIDs are not detected on a host this means the host is missing software and it will be reported in the scorecard output. For a user-defined Patch Report, the missing QIDs list replaces the missing QIDs list set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (missing_qids=), the existing missing QIDs list in the scorecard attributes is not included in the scorecard parameters submitted with the API request. Conditions: a) The missing_qids parameter may be specified only for a Patch Report. b) For a Patch Report, patch_qids or missing_qids must be specified. Both parameters may be specified together.</td>
</tr>
</tbody>
</table>

DTD

&lt;platform API server&gt;/api/2.0/simple_return.dtd
### Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=cancel</td>
<td>(Required) Specifies the report ID of a running report that you want to cancel. The status of the report must be “running”.</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

**Sample - Cancel running report**

```bash
curl -H "X-Requested-With: Curl Sample"
-d "action=cancel&id=1462"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/
```

**DTD**

```xml
<platform API server>/api/2.0/simple_return.dtd
```

### Download Saved Report

/api/2.0/fo/report/

[GET] [POST]

Download a saved report in the user’s account. You can download all report types (map, scan, patch, authentication, scorecard, remediation, compliance). This option is available when the Report Share feature is enabled in the user’s subscription.

You can also view risk scores to Host Based Scan Reports, including TruRisk score, Asset Criticality Score (ACS) and Qualys Detection Score (QDS). These values appear in all report formats, including XML and CSV. You can download reports from the UI or fetch reports using the API.

Downloading a Policy Report in CSV format? When PCRS is enabled for your subscription, we’ll automatically compress large CSV policy reports and you’ll get a Zip file instead of CSV when the report is greater than 1GB in size. See Launching and Fetching Compliance Reports in CSV Format for important details.

User permissions - Managers can download any saved report. Unit Managers can download a saved report in their own business unit (reports launched by users in their own business unit). Scanners and Readers can download their own saved report.
Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=fetch</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) Specifies the report ID of a saved report that you want to download. The status of the report must be “finished”.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

Where do I get the report ID?

Run the report list API

API request:

```bash
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE REPORT_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/report/report_list_output.dtd">
<REPORT_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-07-02T15:29:52Z</DATETIME>
    <REPORT_LIST>
      <REPORT>
        <ID>7592049</ID>
        <TITLE><![CDATA[FIXED Vuln Report]]></TITLE>
        <TYPE>Scan</TYPE>
        <USER_LOGIN>acme_ur15</USER_LOGIN>
        <LAUNCH_DATETIME>2018-07-02T14:52:45Z</LAUNCH_DATETIME>
        <OUTPUT_FORMAT>HTML</OUTPUT_FORMAT>
        <SIZE>-</SIZE>
        <STATUS>
          <STATE>Running</STATE>
          <MESSAGE><![CDATA[Rendering...]]></MESSAGE>
          <PERCENT>80</PERCENT>
        </STATUS>
        <EXPIRATION_DATETIME>2018-07-30T14:52:48Z</EXPIRATION_DATETIME>
      </REPORT>
      ...
    </REPORT_LIST>
  </RESPONSE>
</REPORT_LIST_OUTPUT>
```
Another option - go to the user interface
Within the user interface find the report you want to download (go to Reports > Reports) then choose View Report. In the Report Information window, at the top you’ll see the ID in the window URL after id= like this:


Sample - Download report

curl -H "X-Requested-With: Curl Sample" -b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/report/?action=fetch&id=1462"

Sample - Download Host Based Scan Report in CSV Format

In this sample, we’re downloading a Host Based Scan Report in CSV format. You’ll see the new column headers "QDS", "ARS" and "ACS".

API request:


CSV output:

"Sample Report","05/24/2022 at 18:17:24 (GMT-0800)"
"Qualys","919 E Hillsdale Blvd","Foster City","California","United States of America","94404"
"Joe User","joe_user","Manager"
"IP","DNS","NetBIOS","QG Host ID","IP Interfaces","Tracking Method","OS" "IP Status","QID","Title","Vuln Status","Type","Severity","Port","Protocol","FQDN","SSL","First Detected","Last Detected","Times Detected","Date Last Fixed","First Reopened","Last Reopened","Times Reopened","CVE ID","Vendor Reference","Bugtraq ID","Threat","Impact","Solution","Exploitability","Associated Malware","Results","PCI Vuln","Ticket State","Instance","OS CPE","Category","Associated Ags","Cloud Provider","Cloud Provider Service","Cloud Service","Cloud Resource ID","Cloud Resource Type","Cloud Account","Cloud Image ID","Cloud Resource Metadata","EC2 Instance ID","Public Hostname","Image ID","VPC ID","Instance State","Private Hostname","Instance Type","Account ID","Region Code","Subnet ID","Host ID","Asset ID","QDS","ARS","TRURISK SCORE","ACS"

10.20.30.40","10-20-30-40.bogus.tld",,,,"DNS",,host scanned, found vuln","Microsoft Internet Explorer TABLE Status Bar URI Obfuscation Weakness","New","Vuln","2","05/24/2022 10:07:23","05/24/2022 10:07:23","1","CVE-2005-4679","11561","Microsoft Internet Explorer is reported prone to a URI obfuscation weakness. The issue presents itself when a HREF tag contains an additional HREF tag contained within a TABLE tag. It is reported that hovering over the link of the second HREF tag will display the hostname address of the first HREF tag in the status bar of Internet Explorer.

This weakness is reported to affect Internet Explorer 6, but other versions may also be affected. Windows XP Service Pack 2 is not reported to be vulnerable."This issue may be leveraged by an attacker to display false information in the status bar of an unsuspecting user, allowing an attacker to present Web pages to users that seem to originate from a trusted location. This may facilitate phishing style attacks. Other attacks may also be possible."This vulnerability is not exploitable with Windows XP Service Pack 2. There are no solutions available at this time for Windows 2000 or Windows XP Service Pack 1."yes","Internet Explorer",[]","2685870","14617851","28","104","4"

Sample - Download Host Based Scan Report in XML Format
In this sample, we’re downloading a Host Based Scan Report in XML format. You’ll see <ARS> and <ACS> as part of Host details, and you’ll see <QDS> as part of Vuln Info.

API request:

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ASSET_DATA_REPORT SYSTEM "https://qualysapi.qualys.com/asset_data_report.dtd">
<ASSET_DATA_REPORT>
<HEADER>
  <COMPANY>
    <![CDATA[ Qualys ]]>  
  </COMPANY>
  <USERNAME>joe_user</USERNAME>
  <GENERATION_DATETIME>2022-05-24T15:30:56Z</GENERATION_DATETIME>
  <TEMPLATE>
    <![CDATA[ ARS_Report ]]>
  </TEMPLATE>
</HEADER>
<TARGET>
  <USER_IP_LIST>
    <RANGE>
      <START>10.20.30.40</START>
      <END>10.20.30.40</END>
    </RANGE>
  </USER_IP_LIST>
  <COMBINED_IP_LIST>
    <RANGE>
      <START>10.20.30.40</START>
      <END>10.20.30.40</END>
    </RANGE>
  </COMBINED_IP_LIST>
</TARGET>
<RISK_SCORE_SUMMARY>
  <TOTAL_VULNERABILITIES>5</TOTAL_VULNERABILITIES>
  <AVG_SECURITY_RISK>2.2</AVG_SECURITY_RISK>
  <BUSINESS_RISK>10/100</BUSINESS_RISK>
</RISK_SCORE_SUMMARY>
<RISK_SCORE_PER_HOST>
  <HOSTS>
    <HOST>
      <IP>10.20.30.40</IP>
      <TOTAL_VULNERABILITIES>5</TOTAL_VULNERABILITIES>
      <SECURITY_RISK>2.2</SECURITY_RISK>
    </HOST>
  </HOSTS>
</RISK_SCORE_PER_HOST>
<HOST_LIST>
  <HOST>
    <IP>10.20.30.40</IP>
    <TRACKING_METHOD>DNS</TRACKING_METHOD>
    <HOST_ID>2685870</HOST_ID>
    <ASSET_ID>14617851</ASSET_ID>
    <DNS>
      <![CDATA[ 10-20-30-40.bogus.tld ]]>  
    </DNS>
    <ARS>104</ARS>
    <TRURISK_SCORE>104</TRURISK_SCORE>
    <ACS>4</ACS>
    <VULN_INFO_LIST>
      <VULN_INFO>
        <QID id="qid_100027">100027</QID>
        <TYPE>Practice</TYPE>
        <SSL>false</SSL>
      </VULN_INFO>
    </VULN_INFO_LIST>
  </HOST>
</HOST_LIST>
</ASSET_DATA_REPORT>
<FIRST_FOUND>2022-05-24T04:37:23Z</FIRST_FOUND>
<LAST_FOUND>2022-05-24T04:37:23Z</LAST_FOUND>
<TIMES_FOUND>1</TIMES_FOUND>
<VULN_STATUS>New</VULN_STATUS>
<QDS>
  <![CDATA[ 32 ]]> 
</QDS>
</VULN_INFO>

DTD
<platform API server>/asset_data_report.dtd
Delete Saved Report

/api/2.0/fo/report

[POST]

Delete a saved report in the user’s account. This option is available when the Report Share feature is enabled in the user’s subscription.

User permissions - Managers can delete any saved report. Unit Managers can delete a saved report in their own business unit (report launched by users in their own business unit). Scanners and Readers can delete their own saved report.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) Specifies the report ID of a saved report in Report Share that you want to delete. The status of the report must be &quot;finished&quot;.</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

Sample - Delete saved report

curl -H "X-Requested-With: Curl Sample"
-d "action=delete&id=1234"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api; secure" "https://qualysapi.qualys.com/api/2.0/fo/report/"

DTD

<platform API server>/api/2.0/simple_return.dtd
Scheduled Reports List

/api/2.0/fo/schedule/report/ with action=list

[GET] [POST]

List scheduled reports in your account.

Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Optional) Show only 1 scheduled report that has the report ID you specify.</td>
</tr>
<tr>
<td>is_active={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

Sample - List all scheduled reports in account

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/report/?action=list"

DTD

<platform API server>/api/2.0/fo/schedule/report/schedule_report_list_output.dtd
Launch Scheduled Report

/api/2.0/fo/schedule/report/ with action=launch_now

[POST]

Launch a scheduled report now.

Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=launch_now</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) A valid scheduled report ID.</td>
</tr>
</tbody>
</table>

Sample - Launch scheduled report

```
curl -H "X-Requested-With: Curl" -u USERNAME:PASSWORD -X "POST" -d "action=launch_now&id=12345" "https://qualysapi.qualys.com/api/2.0/fo/schedule/report/"
```

DTD

<platform API server>/api/2.0/simple_return.dtd

Asset Search Report

/api/2.0/fo/report/asset/?action=search

[GET] [POST]

Download report on assets you’re interested in.

Input parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=search</td>
<td>(Required)</td>
</tr>
<tr>
<td>output_format={csv</td>
<td>xml}</td>
</tr>
<tr>
<td>tracking_method={value}</td>
<td>(Optional) Show only IP addresses/ranges which have a certain tracking method. Valid values: IP, DNS, NETBIOS, AZURE VM, EC2, AGENT</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional) Use this parameter if you want to include only certain IP addresses in the report. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.10.1-10.10.10.100). One of these parameters must be specified in a request: ips, asset_groups, asset_group_ids, or use_tags.</td>
</tr>
<tr>
<td>ips_network_id={value}</td>
<td>(Optional) The network ID applied on IPs. The default value is ALL.</td>
</tr>
<tr>
<td>asset_group_ids={value}</td>
<td>(Optional) The IDs of asset groups containing the hosts to be included in the asset search report. Multiple IDs are comma separated.</td>
</tr>
<tr>
<td>asset_groups={value}</td>
<td>(Optional) The titles of asset groups containing the hosts to be included in the asset search report. Multiple titles are comma separated.</td>
</tr>
<tr>
<td>assets_in_my_network_only={0</td>
<td>1}</td>
</tr>
<tr>
<td>ec2_instance_status={value}</td>
<td>(Optional) Specify the EC2 instance status to be searched. Possible values: RUNNING, TERMINATED, PENDING, STOPPING, SHUTTING_DOWN, STOPPED. Values are case-sensitive. ec2_instance_status is valid only when tracking_method=EC2 or tracking_method=AGENT is specified. See EC2 search samples</td>
</tr>
<tr>
<td>ec2_instance_id={value}</td>
<td>(Optional) Specify the EC2 instance ID to be searched. See EC2 search samples ec2_instance_id is valid only when ec2_instance_id Modifier is specified</td>
</tr>
<tr>
<td>ec2_instance_id_modifier={value}</td>
<td>(Optional) Show only hosts with ec2_instance_id that is either beginning with, containing, matching, ending with, not empty. See EC2 search samples ec2_instance_id Modifier is valid only when ec2_instance_id is specified</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>azure_vm_state={value}</td>
<td>(Optional) Specify the Azure virtual machine state to be searched. Possible values are: STARTING, RUNNING, STOPPING, STOPPED, DEALLOCATING, DEALLOCATED, UNKNOWN. Values are case-sensitive. azure_vm_state is valid only when tracking_method=AZURE VM or tracking_method=AGENT is specified.</td>
</tr>
<tr>
<td>azure_vm_id={value}</td>
<td>(Optional) Specify the Azure virtual machine ID to be searched. azure_vm_id is valid only when azure_vm_id_modifier is specified.</td>
</tr>
<tr>
<td>azure_vm_id_modifier={value}</td>
<td>(Optional) Show only assets with azure_vm_id that is either: beginning with, containing, matching, ending with, not empty. azure_vm_id_modifier is valid only when azure_vm_id is specified.</td>
</tr>
<tr>
<td>display_ag_titles={0</td>
<td>1}</td>
</tr>
<tr>
<td>ports={value}</td>
<td>(Optional) Shows the hosts that has the specified open ports. One or more ports may be specified. Multiple ports are comma separated. You can specify upto 10 values.</td>
</tr>
<tr>
<td>services={value}</td>
<td>(Optional) Shows the hosts that has the specified services running on it. One or more services may be specified. Multiple services are comma separated. You can specify upto 10 values.</td>
</tr>
<tr>
<td>qids={value}</td>
<td>(Optional) Shows vulnerabilities (QIDs) in the KnowledgeBase applicable to the host. Allows up to 20 values.</td>
</tr>
<tr>
<td>qid_with_text={value}</td>
<td>(Optional) Shows vulnerabilities (QIDs) with the specified text in the KnowledgeBase applicable to the host. qid_with_text is valid only when qids parameter is specified.</td>
</tr>
<tr>
<td>qid_with_modifier={value}</td>
<td>(Optional) Show only hosts with QID that is either: beginning with, containing, matching, ending with. qid_with_modifier is valid only when qid_with_text is specified.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>use_tags=[0</td>
<td>1])</td>
</tr>
<tr>
<td>tag_set_by={id</td>
<td>name}</td>
</tr>
<tr>
<td>tag_include_selector= {any</td>
<td>all}</td>
</tr>
<tr>
<td>tag_exclude_selector= {any</td>
<td>all}</td>
</tr>
<tr>
<td>tag_set_include={value}</td>
<td>(Required when use_tags=1) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>tag_set_exclude={value}</td>
<td>(Optional when use_tags=1) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>first_found_days={value}</td>
<td>(Optional) Specify a number of days along with the first_found_modifier so that the range includes the first found date to be searched for. first_found_days is valid only when first_found_modifier is specified.</td>
</tr>
<tr>
<td>first_found_modifier={within</td>
<td>not within}</td>
</tr>
<tr>
<td>last_vm_scan_days={value}</td>
<td>(Optional) Specify a number of days so that it includes the last vm scan date to be searched for. last_vm_scan_days is valid only when last_vm_scan_modifier is specified.</td>
</tr>
<tr>
<td>last_vm_scan_modifier={within</td>
<td>not within}</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>last_pc_scan_days={value}</td>
<td>(Optional) Specify a number of days so that the specified value along with the modifier forms the date range that includes the last scan date to be searched for.</td>
</tr>
<tr>
<td></td>
<td>This parameter is valid only when the policy compliance module is enabled for the user account.</td>
</tr>
<tr>
<td>last_pc_scan_modifier= {within</td>
<td>not within}</td>
</tr>
<tr>
<td></td>
<td>This parameter is valid only when the policy compliance module is enabled for the user account.</td>
</tr>
<tr>
<td>last_scap_scan_days={value}</td>
<td>(Optional) Specify a number of days so that the specified value along with the modifier forms the date range that includes the last SCAP scan date to be searched for.</td>
</tr>
<tr>
<td></td>
<td>This parameter is valid only when the policy compliance module is enabled for the user account.</td>
</tr>
<tr>
<td>last_scap_scan_modifier= {within</td>
<td>not within}</td>
</tr>
<tr>
<td></td>
<td>This parameter is valid only when the policy compliance module is enabled for the user account.</td>
</tr>
<tr>
<td>dns_name={value}</td>
<td>(Optional) Specify the DNS name of the host that needs to be searched.</td>
</tr>
<tr>
<td></td>
<td>dns_name is valid only when dns_modifier is specified.</td>
</tr>
<tr>
<td>dns_modifier={value}</td>
<td>(Optional) Show only hosts with dns_name that is either: beginning with, containing, matching, ending with, not empty.</td>
</tr>
<tr>
<td></td>
<td>dns_modifier is valid only when dns_name is specified.</td>
</tr>
<tr>
<td>netbios_name={value}</td>
<td>(Optional) Specify the NETBIOS name of the host to be searched.</td>
</tr>
<tr>
<td></td>
<td>netbios_name is valid only when netbios_modifier is specified.</td>
</tr>
<tr>
<td>netbios_modifier={value}</td>
<td>(Optional) Show only hosts with netbios_name that is either: beginning with, containing, matching, ending with, not empty.</td>
</tr>
<tr>
<td></td>
<td>netbios_modifier is valid only when netbios_name is specified.</td>
</tr>
<tr>
<td>os_cpe_name={value}</td>
<td>(Optional) Specify the OS CPE name of the host to be searched.</td>
</tr>
<tr>
<td></td>
<td>os_cpe_name is valid only when os_cpe_name is specified.</td>
</tr>
</tbody>
</table>
Asset Search Report

Sample - Request Asset Search report

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/report/asset/?action=search&output_format=xml&echo_request=1&ips=10.10.10.10-10.10.10.20"

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ASSET_SEARCH_REPORT SYSTEM
"https://qualysapi.qualys.com/asset_search_report_v2.dtd">

<ASSET_SEARCH_REPORT>

<HEADER>
  <REQUEST>
    <DATETIME>2018-06-03T20:21:13Z</DATETIME>
    <USER_LOGIN>john_sm</USER_LOGIN>
  </REQUEST>
</HEADER>

<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/report/asset/</RESOURCE>

<PARAM_LIST>
  <PARAM>
    <KEY>action</KEY>
    <VALUE>search</VALUE>
  </PARAM>
  <PARAM>
    <KEY>output_format</KEY>
    <VALUE>xml</VALUE>
  </PARAM>
  <PARAM>
    <KEY>echo_request</KEY>
    <VALUE>1</VALUE>
  </PARAM>
</PARAM_LIST>

</ASSET_SEARCH_REPORT>
```

Parameter | Description
--- | ---
o_s_cpe_modifer={value} | (Optional) Show only hosts with os cpe_name that is either: beginning with, containing, matching, ending with, not empty.
o_s_cpe_modifer is valid only when os_cpe_name is specified.
o_s_name={value} | (Optional) Specify the operating system name of the host to be searched.
o_s_name is valid only when os_modifier is specified.
o_s_modifier={value} | (Optional) Show only hosts with os_name that is either: beginning with, containing, matching, ending with.
o_s_modifier is valid only when os_name is specified.
REPORTS

Asset Search Report

<PARAM>
  <KEY>ips</KEY>
  <VALUE>10.10.10.10-10.10.10.15</VALUE>
</PARAM>

<PARAM_LIST>
</PARAM_LIST>

<REQUEST>
  <COMPANY>Corsa</COMPANY>
  <USERNAME>John Smith</USERNAME>
  <GENERATION_DATETIME>2018-06-03T20:21:13Z</GENERATION_DATETIME>
  <TOTAL>2</TOTAL>
  <FILTERS>
    <IP_LIST>
      <RANGE>
        <START>10.10.10.10</START>
        <END>10.10.10.15</END>
      </RANGE>
    </IP_LIST>
  </FILTERS>
</HEADER>

<HOST_LIST>
  <HOST>
    <IP><![CDATA[10.10.10.10]]></IP>
    <TRACKING_METHOD>IP address</TRACKING_METHOD>
    <OPERATING_SYSTEM><![CDATA[Linux 2.4-2.6 / Embedded Device / F5 Networks Big-IP]]></OPERATING_SYSTEM>
    <LAST_SCAN_DATE>2018-06-03T09:11:21Z</LAST_SCAN_DATE>
    <FIRST_FOUND_DATE>2018-06-03T07:11:46Z</FIRST_FOUND_DATE>
  </HOST>

  <HOST>
    <IP><![CDATA[10.10.10.11]]></IP>
    <TRACKING_METHOD>IP address</TRACKING_METHOD>
    <DNS><![CDATA[10-10-10-11.bogus.tld]]></DNS>
    <NETBIOS><![CDATA[SYS_10_10_10_11]]></NETBIOS>
    <OPERATING_SYSTEM><![CDATA[Windows 2000 Server Service Pack 4]]></OPERATING_SYSTEM>
    <LAST_SCAN_DATE>2018-06-03T07:12:47Z</LAST_SCAN_DATE>
    <LAST_COMPLIANCE.Scan_DATE>2018-05-13T21:15:01Z</LAST_COMPLIANCE.Scan_DATE>
    <FIRST_FOUND_DATE>2018-05-12T15:16:54Z</FIRST_FOUND_DATE>
  </HOST>
</HOST_LIST>

</ASSET_SEARCH_REPORT>
DTD:
<platform API server>/asset_search_report_v2.dtd

Sample - Asset Search report CSV

CSV output:

----BEGIN_RESPONSE_HEADER_CSV
"Launch Datetime","User Login","Resource","Parameter Name","Parameter Value"
,"","output_format","csv"
,"","echo_request","1"
,"","ips","10.10.10.10-10.10.10.20"
----END_RESPONSE_HEADER_CSV

"Company","UserName","ReportDate","AssetGroups","IPAddresses","DNS Hostname","NetBIOSHostname","TargetTrackingMethod","TargetOperatingSystem","TargetService","TargetPort","TargetQID","QIDTitle","TargetLastScanDate","TargetFirstFoundDate","OSCPE","Tags","TargetComplianceLastScanDate","Total"
"Corsa","John Smith","2018-06-07T22:51:23Z","","10.10.10.10-10.10.10.20",,,,,,,,,,,"2"
"IP","DNSHostname","NetBIOSHostname","OperatingSystem","OSCPE","Port/Service/Default Service","TrackingMethod","LastScanDate","LastComplianceScanDate","First Found","Tags"
"10.10.10.10","Linux 2.4-2.6 / Embedded Device / F5 Networks Big-IP","IP address","2018-06-03T09:11:21Z","2018-06-03T07:11:46Z",
"10.10.10.11","SYS_10_10_10_11","IP address","2018-05-12T15:16:54Z",,

Sample - Asset Search Report in XML output for Azure VM instances

This sample will return the asset search report in XML format. In XML output, you'll see these Azure VM instance specific tags: FILTER_AZURE_VM_ID, FILTER_AZURE_VM_STATE with Azure filter values.

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/report/asset/?action=search&asset_groups=All&azure_vm_id=399af5dc-c32a-4c40-95a5-c6ed0e786430&azure_vm_id_modifier=beginning+with&tracking_method=AZURE+VM&azure_vm_state=RUNNING&output_format=xml"
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE ASSET_SEARCH_REPORT SYSTEM
  "https://qualysapi.qualys.com/asset_search_report_v2.dtd">

<ASSET_SEARCH_REPORT>
  <HEADER>
    <COMPANY><![CDATA[Qualys]]></COMPANY>
    <USERNAME>Patrick Slimmer</USERNAME>
    <GENERATION_DATETIME>2020-06-22T23:24:25Z</GENERATION_DATETIME>
    <TOTAL>1</TOTAL>
    <FILTERS>
      <ASSET_GROUPS>
        <ASSET_GROUP_TITLE><![CDATA[All]]></ASSET_GROUP_TITLE>
      </ASSET_GROUPS>
      <FILTER_AZURE_VM_ID><![CDATA[Beginning With 399af5dc-c32a-4c40-95a5-c6ed0e786430]]></FILTER_AZURE_VM_ID>
      <TRACKING_METHOD><![CDATA[Azure VM]]></TRACKING_METHOD>
      <FILTER_AZURE_VM_STATE><![CDATA[RUNNING]]></FILTER_AZURE_VM_STATE>
    </FILTERS>
  </HEADER>
  <HOST_LIST>
    <HOST>
      <IP><![CDATA[10.4.8.4]]></IP>
      <TRACKING_METHOD>Azure VM</TRACKING_METHOD>
      <CLOUD_PROVIDER>Azure</CLOUD_PROVIDER>
      <CLOUD_SERVICE>VM</CLOUD_SERVICE>
      <CLOUD_RESOURCE_ID><![CDATA[399af5dc-c32a-4c40-95a5-c6ed0e786430]]></CLOUD_RESOURCE_ID>
    </HOST>
  </HOST_LIST>
</ASSET_SEARCH_REPORT>
```

Sample - Search EC2 asset with certain EC2 instance ID

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=search&output_format=xml&tracking_method=EC2&use_tags=1&tag_set_by=name&tag_set_include=useasttag&ec2_instance_id=i-0fb708f985856fa4&ec2_instance_id_modifier=containing" "https://qualysapi.qualys.com/api/2.0/fo/report/asset/
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE ASSET_SEARCH_REPORT SYSTEM
  "https://qualysapi.qualys.com/asset_search_report_v2.dtd">
```
"https://qualysapi.qualys.com/asset_search_report_v2.dtd">
<ASSET_SEARCH_REPORT>
<HEADER>
  <COMPANY><![CDATA[qualys-test]]></COMPANY>
  <USERNAME>qualys_ps</USERNAME>
  <GENERATION_DATETIME>2018-04-11T10:17:32Z</GENERATION_DATETIME>
  <TOTAL>1</TOTAL>
  <FILTERS>
    <ASSET_TAGS>
      <INCLUDED_TAGS scope="any">
        <ASSET_TAG><![CDATA[useasttag]]></ASSET_TAG>
      </INCLUDED_TAGS>
    </ASSET_TAGS>
    <TRACKING_METHOD><![CDATA[EC2]]></TRACKING_METHOD>
  </FILTERS>
</HEADER>
<HOST_LIST>
  <HOST>
    <IP><![CDATA[10.73.188.6]]></IP>
    <HOST_TAGS><![CDATA[EC2, Virginia, agec2, sada-0117-targets, sada-new-0308, useasttag;]]></HOST_TAGS>
    <TRACKING_METHOD>EC2</TRACKING_METHOD>
    <DNS><![CDATA[ip-10-73-188-6.ec2.internal]]></DNS>
    <EC2_INSTANCE_ID><![CDATA[i-0fb70b6f985856fa4]]></EC2_INSTANCE_ID>
    <LAST_SCAN_DATE />  
    <FIRST_FOUND_DATE />  
  </HOST>
</HOST_LIST>

Sample - Search EC2 assets with certain status

Search all EC2 assets which are currently in TERMINATED state and having instance ID i-0b121b9211d7e25cb.

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d 
"action=search&output_format=xml&tracking_method=EC2&use_tags=1&tag_set_by=name&tag_set_include=useasttag&ec2_instance_status=TERMINATED&ec2_instance_id=i-0b121b9211d7e25cb&ec2_instance_id_modifier=containing"
"https://qualysapi.qualys.com/api/2.0/fo/report/asset/

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
Sample - Search assets with SCAP scan performed

API request:

curl -u "username:password" -H "X-Requested-With:
"action=search&output_format=xml&asset_groups=Winodws+7+Scap&last_ scap_scan_days=300&last_scap_scan Modifier=within
"https://qualysapi.qualys.com/api/2.0/fo/report/asset/

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ASSET_SEARCH_REPORT SYSTEM
"https://qualysapi.qualys.com/asset_search_report_v2.dtd">
<ASSET_SEARCH_REPORT>
<HEADER>
  <COMPANY><![CDATA[qualys-test]]></COMPANY>
  <USERNAME>sada-customer customer</USERNAME>
  <GENERATION_DATETIME>2018-04-11T10:49:05Z</GENERATION_DATETIME>
  <TOTAL>1</TOTAL>
  <FILTERS>
    <ASSET_TAGS>
      <INCLUDED_TAGS scope="any">
        <ASSET_TAG><![CDATA[useasttag]]></ASSET_TAG>
      </INCLUDED_TAGS>
    </ASSET_TAGS>
    <TRACKING_METHOD><![CDATA[EC2]]></TRACKING_METHOD>
  </FILTERS>
</HEADER>
<HOST_LIST>
  <HOST>
    <IP><![CDATA[10.90.2.175]]></IP>
    <HOST_TAGS><![CDATA[EC2, Virginia, por-6586, sada-0117- targets, sada-new-0308, useasttag;]]></HOST_TAGS>
  </HOST>
</HOST_LIST>
<HEADER>
  <COMPANY><![CDATA[qualys]]></COMPANY>
  <USERNAME>POC Manager</USERNAME>
  <GENERATION_DATETIME>2018-11-06T00:42:13Z</GENERATION_DATETIME>
  <TOTAL>26</TOTAL>
  <FILTERS>
    <ASSET_GROUPS>
      <ASSET_GROUP_TITLE><![CDATA[Winodws 7 Scap]]></ASSET_GROUP_TITLE>
    </ASSET_GROUPS>
    <FILTER_LAST_SCAP_SCAN_DATE><![CDATA[Within 300]]></FILTER_LAST_SCAP_SCAN_DATE>
  </FILTERS>
</HEADER>

<HOST_LIST>
  <HOST>
    <IP><![CDATA[10.10.10.10]]></IP>
    <TRACKING_METHOD>IP address</TRACKING_METHOD>
    <DNS><![CDATA[bridge.qualys.com]]></DNS>
    <NETBIOS><![CDATA[WIN7-10-10]]></NETBIOS>
    <OPERATING_SYSTEM><![CDATA[Windows 7 Ultimate 64 bit Edition Service Pack 1]]></OPERATING_SYSTEM>
    <OS_CPE><![CDATA[cpe:/o:microsoft:windows_7::sp1:x64-ultimate:]]></OS_CPE>
    <LAST_SCAP_SCAN_DATE>2018-08-28T10:57:06Z</LAST_SCAP_SCAN_DATE>
    <FIRST_FOUND_DATE>2018-04-03T23:18:26Z</FIRST_FOUND_DATE>
  </HOST>
</HOST_LIST>
VM Report Templates

The Report Template API is used to manage report templates and their settings in the user’s subscription.

API Support for Report Templates

Scan Template
PCI Scan Template
Patch Template
Map Template

API Support for Report Templates

You can now use APIs to create custom reports with views on your scan results and the current vulnerabilities on your hosts. Use various report templates provided by Qualys as a starting point.

APIs are now available to perform various actions on templates for the following report types: Scan Template, PCI Scan Template, Patch Template, Map Template

The Report Template API allows users to perform the following actions.

<table>
<thead>
<tr>
<th>Action</th>
<th>Supported Access Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>POST</td>
<td>Create a report template. A unique template ID is generated for the new template.</td>
</tr>
<tr>
<td>Update</td>
<td>PUT</td>
<td>Update an existing report template.</td>
</tr>
<tr>
<td>Delete</td>
<td>POST</td>
<td>Delete an existing report template.</td>
</tr>
<tr>
<td>Export</td>
<td>GET</td>
<td>Export a specific report template based on the template ID, or all templates for the report type.</td>
</tr>
</tbody>
</table>

Once you have your template the way you want you can run reports using the templates using the Report API /api/2.0/fo/report.
Scan Template

/api/2.0/fo/report/template/scan/

Perform actions such as create, update, delete and export on the Scan Template.

Scan Template Request
A summary of API Endpoint URLs is provided below.

<table>
<thead>
<tr>
<th>Action</th>
<th>API Endpoint /required parameters</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Scan Template</td>
<td>&lt;base_url&gt;/api/2.0/fo/report/template/scan/ Required parameters: action=create report_format=xml</td>
<td>POST</td>
</tr>
<tr>
<td>Update Scan Template</td>
<td>&lt;base_url&gt;/api/2.0/fo/report/template/scan/ Required parameters: template_id={value} action=update report_format=xml</td>
<td>PUT</td>
</tr>
<tr>
<td>Delete Scan Template</td>
<td>&lt;base_url&gt;/api/2.0/fo/report/template/scan/ Required parameters: template_id={value} action=delete</td>
<td>POST</td>
</tr>
<tr>
<td>Export Scan Template</td>
<td>&lt;base_url&gt;/api/2.0/fo/report/template/scan/ Required parameters: action=export report_format=xml Optional parameter: template_id={value} When unspecified all templates for the report type get exported.</td>
<td>GET</td>
</tr>
</tbody>
</table>

Scan Template settings
These parameters (all are optional) are used for a create or update request to define scan template settings. When creating a new template the default value is shown in bold where applicable.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title={value}</td>
<td>A string value for the title. Length is maximum 64 characters.</td>
</tr>
<tr>
<td>owner={value}</td>
<td>Username of the owner of this template. Validity of the owner to create reports is based on the user role or business unit. See About template owner.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Target</td>
<td>What target assets to include in the report.</td>
</tr>
<tr>
<td>scan_selection=HostBased</td>
<td>Specify HostBased for Host Based Findings (default for new template) or ScanBased for Scan Based Findings. Choosing Host Based Findings allows you to report on the latest vulnerability data from all of your scans. Choosing Scan Based Findings allows you to run a report based on saved scan results.</td>
</tr>
<tr>
<td></td>
<td><em>Note: This parameter is required only if scan_selection=HostBased.</em></td>
</tr>
<tr>
<td>include_trending=1</td>
<td>Specify 1 to include trending. Choose a timeframe (daily, weekly or monthly) to analyze the vulnerability status for the timeframe selected. This parameter is required only if scan_selection=HostBased.</td>
</tr>
<tr>
<td>limit_timeframe=1</td>
<td>Specify 1 to only include scan results from the specified time frame. This ensures that only vulnerability information gathered in the timeframe that you've specified is included in the report. If unspecified, vulnerability information for hosts that were last scanned prior to the report timeframe may be included. This parameter is required only if scan_selection=HostBased.</td>
</tr>
<tr>
<td>selection_type=day</td>
<td>Specify whether to include trending information for number of weeks, days or months or a specific date. Specifying none will create a report without any trending information included. Specifying scans will include trending information for the last two detections. This parameter is required only if scan_selection=HostBased.</td>
</tr>
<tr>
<td></td>
<td><em>Note: This parameter is required only if scan_selection=HostBased.</em></td>
</tr>
<tr>
<td>selection_range=value</td>
<td>Specify the range for the selection type. Specify a number of units (1</td>
</tr>
<tr>
<td>asset_groups=value</td>
<td>Specify the name of the asset group(s) to report on. Multiple asset groups are comma separated. We'll report on all the IPs in the asset groups. This parameter is required only if scan_selection=HostBased.</td>
</tr>
<tr>
<td>asset_group_ids=value</td>
<td>Specify the ID of the asset group(s) to report on. Multiple asset group IDs are comma separated. We'll report on all the IPs in the asset groups. This parameter is required only if scan_selection=HostBased.</td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
`network={value}` | (Valid only when the Networks feature is enabled for your account.) A network name containing the IPs to include. For a new template the default network is Global Default Network.

`ips={value}` | Specify the IPs or IP ranges to report on. Multiple IPs or IP ranges are comma separated. This parameter is required only if `scan_selection=HostBased`.

`tag_set_by={name|id}` | Specify the name of the tags or the ID of the tags for the hosts you want to report on. Multiple tag names or tag IDs are comma separated.

`tag_include_selector={ALL|ANY}` | Specify ALL to match all the asset tags for the hosts you want to report on (This is an AND operation). Specifying ANY will match any of the assets tags (This is an OR operation). This parameter is required only if `scan_selection=HostBased`.

`tag_set_include={value}` | Specify asset tags for the hosts you want to report on. We'll find the hosts in your account that match your tag selection and include them in the report. Multiple tags can be provided using comma separated values. This parameter is required only if `scan_selection=HostBased`.

`tag_exclude_selector={ALL|ANY}` | Specify ALL to match all the asset tags for the hosts you want to do not want to report on (This is an AND operation). Specifying ANY will match any of the assets tags (This is an OR operation). This parameter is required only if `scan_selection=HostBased`.

`tag_set_exclude={value}` | Specify asset tags for the hosts you do not want to report on. We'll find the hosts in your account that match your tag selection and exclude them from the report. Multiple tags can be provided using comma separated values. This parameter is required only if `scan_selection=HostBased`.

`host_with_cloud_agents={all|scan|agent}` | What host findings to include in the report when CA module is enabled. Your options are: all - All data scan - Scan data, i.e. include findings from scans that didn't use Agentless Tracking agent - Agent data, i.e. include findings from the agent when merging is enabled (i.e. Show unified view hosts option in UI under Users > Setup > Cloud Agent Setup)
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>display_text_summary={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_business_risk={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_vuln_over_time={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_status={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_potential_status={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_severity={0</td>
<td>1}</td>
</tr>
<tr>
<td>Display</td>
<td>Display options such as graphs amount of detail.</td>
</tr>
<tr>
<td>graph_potential_severity={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_ig_severity={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_top_categories={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_top_vulns={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_os={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_services={0</td>
<td>1}</td>
</tr>
<tr>
<td>graph_top_ports={0</td>
<td>1}</td>
</tr>
<tr>
<td>display_custom_footer={0</td>
<td>1}</td>
</tr>
<tr>
<td>display_custom_footer_text={value}</td>
<td>Specify custom text like a disclosure statement or data classification (e.g., Public, Confidential). The text you enter will appear in all reports generated from this template, except reports in XML and CSV formats. Length is maximum 4000 characters.</td>
</tr>
<tr>
<td>sort_by=[[host</td>
<td>vuln</td>
</tr>
<tr>
<td>cvss={all</td>
<td>cvssv2</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>host_details=[0</td>
<td>1]</td>
</tr>
<tr>
<td>metadata_ec2_instances=[0</td>
<td>1]</td>
</tr>
<tr>
<td>cloud_provider_metadata=[0</td>
<td>1]</td>
</tr>
<tr>
<td>qualys_system_ids=[0</td>
<td>1]</td>
</tr>
<tr>
<td>include_text_summary=[0</td>
<td>1]</td>
</tr>
<tr>
<td>include_vuln_details=[0</td>
<td>1]</td>
</tr>
<tr>
<td>include_vuln_details_threat=[0</td>
<td>1]</td>
</tr>
<tr>
<td>include_vuln_details_impact=[0</td>
<td>1]</td>
</tr>
<tr>
<td>include_vuln_details_solution=[0</td>
<td>1]</td>
</tr>
<tr>
<td>include_vuln_details_vpatch=[0</td>
<td>1]</td>
</tr>
<tr>
<td>include_vuln_details_compliance=[0</td>
<td>1]</td>
</tr>
<tr>
<td>include_vuln_details_exploit=[0</td>
<td>1]</td>
</tr>
<tr>
<td>include_vuln_details_malware=[0</td>
<td>1]</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_vuln_details_results=0</td>
<td>1</td>
</tr>
<tr>
<td>include_vuln_details_reopen=0</td>
<td>1</td>
</tr>
<tr>
<td>include_vuln_details_appended=0</td>
<td>1</td>
</tr>
<tr>
<td>include_trurisk_details=0</td>
<td>1</td>
</tr>
<tr>
<td>exclude_account_id=0</td>
<td>1</td>
</tr>
<tr>
<td>selective_vulns=complete</td>
<td>custom</td>
</tr>
<tr>
<td>search_list_ids=value</td>
<td>Specify search list ID or QID. Multiple search list IDs or QIDs can be provided using values separated by a comma. This parameter is required only if selective_vulns=custom.</td>
</tr>
<tr>
<td>exclude_qid_option=0</td>
<td>1</td>
</tr>
<tr>
<td>exclude_search_list_ids=value</td>
<td>Specify QID to be excluded from the report. Multiple QIDs can be provided using values separated by a comma. This parameter is required only if exclude_qid_option=1.</td>
</tr>
<tr>
<td>included_os=value</td>
<td>Specify the operating system name to filter hosts. For example, to only report on Linux hosts make sure you provide the operating system name for Linux. Multiple operating system names can be provided using values separated by a comma. Specify ALL to include all operating systems. See <a href="#">Identified OS</a>.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>status_new={0</td>
<td>1}</td>
</tr>
<tr>
<td>status_active={0</td>
<td>1}</td>
</tr>
<tr>
<td>status_reopen={0</td>
<td>1}</td>
</tr>
<tr>
<td>status_fixed={0</td>
<td>1}</td>
</tr>
<tr>
<td>vuln_active={0</td>
<td>1}</td>
</tr>
<tr>
<td>vuln_disabled={1</td>
<td>1}</td>
</tr>
<tr>
<td>vuln_ignored={0</td>
<td>1}</td>
</tr>
<tr>
<td>potential_active={0</td>
<td>1}</td>
</tr>
<tr>
<td>potential_disabled={0</td>
<td>1}</td>
</tr>
<tr>
<td>potential_ignored={0</td>
<td>1}</td>
</tr>
<tr>
<td>ig_active={0</td>
<td>1}</td>
</tr>
<tr>
<td>ig_disabled={0</td>
<td>1}</td>
</tr>
<tr>
<td>ig_ignored={0</td>
<td>1}</td>
</tr>
<tr>
<td>display_non_running_kernels={0</td>
<td>1}</td>
</tr>
<tr>
<td>exclude_non_running_kernels={0</td>
<td>1}</td>
</tr>
<tr>
<td>exclude_non_running_services={0</td>
<td>1}</td>
</tr>
<tr>
<td>exclude_qids_not_exploitable_due_to_configuration={0</td>
<td>1}</td>
</tr>
<tr>
<td>exclude_superceded_patches={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>categories_list={value}</td>
<td>Specify the category name to filter hosts in your report based on various categories. For example, if you're only interested in Windows vulnerabilities make sure you provide the category name for Windows. Multiple category names can be provided using values separated by a comma. Specify ALL to include all categories. See Categories.</td>
</tr>
<tr>
<td>Services and Ports</td>
<td>Services and ports to include in report.</td>
</tr>
<tr>
<td>required_services={value}</td>
<td>Specify the name of a required service. Multiple service names can be provided using values separated by a comma. We'll report QID: 38228 (when a required service is NOT detected). See Identified Services.</td>
</tr>
<tr>
<td>unauthorized_services={value}</td>
<td>Specify the name of an unauthorized service. Multiple service names can be provided using values separated by a comma. We'll report QID: 38175 (when an unauthorized service is detected). See Identified Services.</td>
</tr>
<tr>
<td>required_ports={value}</td>
<td>Specify required ports. Multiple ports can be provided using values separated by a comma. We'll report QID: 82051 (when a required port is NOT detected).</td>
</tr>
<tr>
<td>unauthorized_ports={value}</td>
<td>Specify unauthorized ports. Multiple ports can be provided using values separated by a comma. We'll report QID: 82043 (when an unauthorized port is detected).</td>
</tr>
<tr>
<td>User Access</td>
<td>Control user access to template and reports generated from template.</td>
</tr>
<tr>
<td>global={0</td>
<td>1}</td>
</tr>
<tr>
<td>report_access_users={value}</td>
<td>Specify the username to share the report with a user who wouldn't already have access to the report. Multiple usernames can be provided using values separated by a comma. Each user you add will be able to view reports generated from this template even if they don't have access to the IPs in the report.</td>
</tr>
</tbody>
</table>
Cloud Asset Metadata Fields in CSV Format

See the table below to know which cloud asset metadata columns will appear in your CSV reports based on your report template settings. Columns will appear in the order shown.

<table>
<thead>
<tr>
<th>Legacy EC2/Azure Fields</th>
<th>Cloud Provider Metadata Fields</th>
<th>All Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC2 Instance ID</td>
<td>Cloud Provider</td>
<td>Cloud Provider</td>
</tr>
<tr>
<td>Public Hostname</td>
<td>Cloud Provider Service</td>
<td>Cloud Provider Service</td>
</tr>
<tr>
<td>Image ID</td>
<td>Cloud Service</td>
<td>Cloud Service</td>
</tr>
<tr>
<td>VPC ID</td>
<td>Cloud Resource ID</td>
<td>Cloud Resource ID</td>
</tr>
<tr>
<td>Instance State</td>
<td>Cloud Resource Type</td>
<td>Cloud Resource Type</td>
</tr>
<tr>
<td>Private Hostname</td>
<td>Cloud Account</td>
<td>Cloud Account</td>
</tr>
<tr>
<td>Instance Type</td>
<td>Cloud Image ID</td>
<td>Cloud Image ID</td>
</tr>
<tr>
<td>Account ID</td>
<td>Cloud Resource Metadata</td>
<td>Cloud Resource Metadata</td>
</tr>
<tr>
<td>Region Code</td>
<td>EC2 Instance ID</td>
<td>EC2 Instance ID</td>
</tr>
<tr>
<td>Subnet ID</td>
<td>Public Hostname</td>
<td>Public Hostname</td>
</tr>
<tr>
<td></td>
<td>Image ID</td>
<td>Image ID</td>
</tr>
<tr>
<td></td>
<td>VPC ID</td>
<td>VPC ID</td>
</tr>
<tr>
<td></td>
<td>Instance State</td>
<td>Instance State</td>
</tr>
<tr>
<td></td>
<td>Private Hostname</td>
<td>Private Hostname</td>
</tr>
<tr>
<td></td>
<td>Instance Type</td>
<td>Instance Type</td>
</tr>
<tr>
<td></td>
<td>Account ID</td>
<td>Account ID</td>
</tr>
<tr>
<td></td>
<td>Region Code</td>
<td>Region Code</td>
</tr>
<tr>
<td></td>
<td>Subnet ID</td>
<td>Subnet ID</td>
</tr>
</tbody>
</table>

Important note about the Legacy EC2/Azure Fields in CSV

These fields were originally introduced for AWS cloud assets and will be populated with metadata for your AWS EC2 assets.

For Azure and GCP assets, all Legacy EC2/Azure columns will appear blank in the CSV report, except for the EC2 Instance ID column. We will continue to populate the EC2 Instance ID column for all cloud assets (AWS, Azure, GCP). The EC2 Instance ID column is replaced by Cloud Resource ID, and will be deprecated in a future release.

Cloud Asset Metadata Fields in XML Format

See the table below to know which cloud asset metadata tags will appear in your XML reports based on your report template settings.

<table>
<thead>
<tr>
<th>Cloud Provider</th>
<th>Legacy EC2/Azure Fields</th>
<th>Cloud Provider Metadata Fields</th>
<th>All Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS</td>
<td>CLOUD_PROVIDER</td>
<td>CLOUD_PROVIDER</td>
<td>CLOUD_PROVIDER</td>
</tr>
<tr>
<td></td>
<td>CLOUD_PROVIDER_SERVICE</td>
<td>CLOUD_PROVIDER_SERVICE</td>
<td>CLOUD_PROVIDER SERVICE</td>
</tr>
<tr>
<td></td>
<td>CLOUD_SERVICE</td>
<td>CLOUD_SERVICE</td>
<td>CLOUD_SERVICE</td>
</tr>
<tr>
<td></td>
<td>CLOUD_RESOURCE_ID</td>
<td>CLOUD_RESOURCE_TYPE</td>
<td>CLOUD_RESOURCE_TYPE</td>
</tr>
<tr>
<td></td>
<td>CLOUD_ACCOUNT</td>
<td>CLOUD_RESOURCE_ID</td>
<td>CLOUD_RESOURCE_ID</td>
</tr>
<tr>
<td></td>
<td>EC2_INSTANCE_ID</td>
<td>CLOUD_IMAGE_ID</td>
<td>CLOUD_IMAGE_ID</td>
</tr>
<tr>
<td></td>
<td>EC2_INFO</td>
<td>CLOUD_RESOURCE_METADATA</td>
<td>CLOUD_Resource_METADATA</td>
</tr>
</tbody>
</table>
EC2_INFO includes: PUBLIC_DNS_NAME, IMAGE_ID, VPC_ID, INSTANCE_STATE,
PRIVATE_DNS_NAME, INSTANCE_TYPE, ACCOUNT_ID, REGION_CODE, SUBNET_ID

AZURE_VM_INFO includes: PUBLIC_IP_ADDRESS, IMAGE_OFFER, IMAGE_VERSION,
SUBNET, VM_STATE, PRIVATE_IP_ADDRESS, SIZE, SUBSCRIPTION_ID, LOCATION,
RESOURCE_GROUP_NAME

CLOUD_RESOURCE_METADATA for AWS includes: INSTANCE_ID, PUBLIC_DNS_NAME,
PUBLIC_IP_ADDRESS, PRIVATE_IP_ADDRESS, IMAGE_ID, SPOT_INSTANCE,
AVAILABILITY_ZONE, VPC_ID, GROUP_ID, GROUP_NAME, LOCAL_HOSTNAME,
INSTANCE_STATE, PRIVATE_DNS_NAME, INSTANCE_TYPE, ACCOUNT_ID, REGION_CODE,
SUBNET_ID, RESERVATION_ID, MAC_ADDRESS

CLOUD_RESOURCE_METADATA for Azure includes: VM_ID, VM_NAME, PLATFORM,
PUBLIC_IP_ADDRESS, IMAGE_OFFER, IMAGE_PUBLISHER, IMAGE_VERSION, SUBNET,
VM_STATE, PRIVATE_IP_ADDRESS, SIZE, SUBSCRIPTION_ID, LOCATION,
RESOURCE_GROUP_NAME, MAC_ADDRESS

CLOUD_RESOURCE_METADATA for GCP includes: INSTANCE_ID, HOST_NAME,
MACHINE_TYPE, MACHINE_STATE, PROJECT_ID, PUBLIC_IP_ADDRESS, VPC_NETWORK,
ZONE, PRIVATE_IP_ADDRESS, MAC_ADDRESS

DTD
<platform API server>/api/2.0/fo/report/template/scan/reporttemplate_info.dtd

Sample - Create scan template

API request:
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -H
"Content-type: text/xml" --data-binary @scan_export.xml
"https://qualysapi.qualys.com/api/2.0/fo/report/template/scan/?action=create&report_format=xml"

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
Sample - Update Scan template

API request:
```
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2017-04-04T10:52:34Z</DATETIME>
    <CODE>Scan Report Template Updated Successfully [8209]</CODE>
    <TEXT></TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Delete Scan template

API request:
```
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2017-04-04T10:54:37Z</DATETIME>
    <CODE>Scan Report Template(s) Deleted Successfully</CODE>
    <TEXT></TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Export Scan template

Exports the report template based on the template ID. When the template ID is not specified, exports all templates for the report type.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl"
"https://qualysapi.qualys.com/api/2.0/fo/report/template/scan/?action=export&template_id=89470&report_format=xml"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE REPORTTEMPLATE SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/report/template/scan/scanreporttemplate_info.dtd">
<REPORTTEMPLATE>
  <SCANTEMPLATE>
    <TITLE>
      <INFO key="title"> <![CDATA[Scan-Report-To-Create-Do not Change]]> </INFO>
    </TITLE>
    <TARGET>
      <INFO key="scan_selection"> <![CDATA[HostBased]]> </INFO>
      <INFO key="include_trending"> <![CDATA[1]]> </INFO>
      <INFO key="selection_type"> <![CDATA[days]]> </INFO>
      <INFO key="selection_range"> <![CDATA[5]]> </INFO>
      <INFO key="limit_timeframe"> <![CDATA[1]]> </INFO>
      <INFO key="asset_groups"> <![CDATA[PBPS-Targets]]> </INFO>
      <INFO key="tag_set_by"> <![CDATA[id]]> </INFO>
      <INFO key="tag_set_include"> <![CDATA[8644659]]> </INFO>
      <INFO key="tag_set_exclude"> <![CDATA[8262228]]> </INFO>
      <INFO key="tag_include_selector"> <![CDATA[ALL]]> </INFO>
      <INFO key="tag_exclude_selector"> <![CDATA[ALL]]> </INFO>
      <INFO key="network"> <![CDATA[-100]]> </INFO>
      <INFO key="ips"> <![CDATA[10.10.0.1,10.10.0.5]]> </INFO>
      <INFO key="host_with_cloud_agents"> <![CDATA[all]]> </INFO>
    </TARGET>
  <DISPLAY>
```
<INFO key="search_list_ids"><![CDATA[]]}</INFO>
<INFO key="exclude_qid_option"><![CDATA[1]]></INFO>
<INFO key="exclude_search_list_ids"><![CDATA[]]]></INFO>
<INFO key="included_os"><![CDATA[ALL]]></INFO>
<INFO key="status_new"><![CDATA[1]]></INFO>
<INFO key="status_active"><![CDATA[1]]></INFO>
<INFO key="status_reopen"><![CDATA[1]]></INFO>
<INFO key="status_fixed"><![CDATA[1]]></INFO>
<INFO key="vuln_active"><![CDATA[1]]></INFO>
<INFO key="vuln_disabled"><![CDATA[1]]></INFO>
<INFO key="vuln_ignored"><![CDATA[0]]></INFO>
<INFO key="potential_active"><![CDATA[1]]></INFO>
<INFO key="potential_disabled"><![CDATA[1]]></INFO>
<INFO key="potential_ignored"><![CDATA[1]]></INFO>
<INFO key="ig_active"><![CDATA[1]]></INFO>
<INFO key="ig_disabled"><![CDATA[1]]></INFO>
<INFO key="ig_ignored"><![CDATA[0]]></INFO>
<INFO key="display_non_running_kernels"><![CDATA[1]]></INFO>
<INFO key="exclude_non_running_kernels"><![CDATA[0]]></INFO>
<INFO key="exclude_non_running_services"><![CDATA[1]]></INFO>
<INFO key="exclude_superceded_patches"><![CDATA[1]]></INFO>
<INFO key="exclude_qids_not_exploitable_due_to_configuration"><![CDATA[1]]></INFO>
<INFO key="categories_list"><![CDATA[ALL]]></INFO>
</FILTER>
<SERVICESPORTS>
<INFO key="required_services"><![CDATA[ActiveSync,akak trojan,Apple Airport Management,Applix TM1 Server]]></INFO>
<INFO key="unauthorized_services"><![CDATA[aml,Arkeiad Network Backup,auth]]></INFO>
<INFO key="services_info"><![CDATA[aml,Arkeiad Network Backup,auth]]></INFO>
<INFO key="required_ports"><![CDATA[12]]></INFO>
<INFO key="unauthorized_ports"><![CDATA[21]]></INFO>
</SERVICESPORTS>
<USERACCESS>
<INFO key="report_access_users"><![CDATA[start_rm2,start_su]]></INFO>
<INFO key="global"><![CDATA[1]]></INFO>
</USERACCESS>
</SCANTEMPLATE>
PCI Scan Template

/api/2.0/fo/report/template/pciscan/

Perform actions such as create, update, delete and export on the PCI Scan Template.

**PCI Scan Template Request**
A summary of API Endpoint URLs is provided below.

<table>
<thead>
<tr>
<th>Action</th>
<th>API Endpoint /required parameters</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create PCI Scan Template</td>
<td><code>&lt;base_url&gt;/api/2.0/fo/report/template/pciscan/</code>&lt;br&gt;<strong>Required parameters:</strong> action=create&lt;br&gt;report_format=xml</td>
<td>POST</td>
</tr>
<tr>
<td>Update PCI Scan Template</td>
<td><code>&lt;base_url&gt;/api/2.0/fo/report/template/pciscan/</code>&lt;br&gt;<strong>Required parameters:</strong> template_id={value}&lt;br&gt;action=update&lt;br&gt;report_format=xml</td>
<td>PUT</td>
</tr>
<tr>
<td>Delete PCI Scan Template</td>
<td><code>&lt;base_url&gt;/api/2.0/fo/report/template/pciscan/</code>&lt;br&gt;<strong>Required parameters:</strong> template_id={value}&lt;br&gt;action=delete</td>
<td>POST</td>
</tr>
<tr>
<td>Export PCI Scan Template</td>
<td><code>&lt;base_url&gt;/api/2.0/fo/report/template/pciscan/</code>&lt;br&gt;<strong>Required parameters:</strong> action=export&lt;br&gt;report_format=xml&lt;br&gt;<strong>Optional parameter:</strong> template_id={value}&lt;br&gt;When unspecified all templates for the report type get exported.</td>
<td>GET</td>
</tr>
</tbody>
</table>
PCI Scan Template settings

Go to Scan Template settings. The same parameters used to define PCI Scan Template settings. All parameters (all are optional).

In addition the following parameters are used for PCI Risk Ranking.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>custom_pci_ranking={0</td>
<td>1]}</td>
</tr>
<tr>
<td>customized_ranking_medium_from={0</td>
<td>1</td>
</tr>
<tr>
<td>customized_ranking_high_from={0</td>
<td>1</td>
</tr>
<tr>
<td>customized_ranking_comments={value}</td>
<td>When custom PCI risk ranking is enabled, a comment on the custom ranking is required. Enter any string up to 400 characters.</td>
</tr>
<tr>
<td>customized_ranking_qid_searchlist_comments={&lt;search list id1/name1</td>
<td>SEVERITY</td>
</tr>
</tbody>
</table>

DTD
<platform API server>/api/2.0/fo/report/template/pciscan/pciscanreporttemplate_info.dtd

Samples
Refer to Scan template examples for create, update, delete and export sample requests. Requests and outputs for PCI Scan template are similar.
Patch Template

/\api/2.0/fo/report/template/patch/

Perform actions such as create, update, delete and export on the Patch Template.

Patch Template Request

A summary of API Endpoint URLs is provided below.

<table>
<thead>
<tr>
<th>Action</th>
<th>API Endpoint /required parameters</th>
<th>Method</th>
</tr>
</thead>
</table>
| Create Patch Template | <base_url>/api/2.0/fo/report/template/patch/  
Required parameters:  
action=create  
report_format=xml | POST   |
| Update Patch Template | <base_url>/api/2.0/fo/report/template/patch/  
Required parameters:  
template_id={value}  
action=update  
report_format=xml | PUT    |
| Delete Patch Template | <base_url>/api/2.0/fo/report/template/patch/  
Required parameters:  
template_id={value}  
action=delete        | POST   |
| Export Patch Template | <base_url>/api/2.0/fo/report/template/patch/  
Required parameters:  
action=export  
report_format=xml | GET    |

Optional parameter:  
template_id={value}  
When unspecified all templates for the report type get exported.

Patch Template settings

These parameters (all are optional) are used for a create or update request to define Patch template settings. When creating a new template the default value is shown in bold where applicable.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title={value}</td>
<td>A string value for the title. Length is maximum 64 characters.</td>
</tr>
</tbody>
</table>
| owner={value} | Username of the owner of this template.  
Validity of the owner to create reports is based on the user role or business unit. See About template owner. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>patch_evaluation=</td>
<td>Specify classic to choose Classic patch evaluation or specify qidbased to choose QID based patch evaluation.</td>
</tr>
<tr>
<td>asset_groups</td>
<td>Asset groups to include in the report. Multiple asset groups are comma separated.</td>
</tr>
<tr>
<td>asset_group_ids={value}</td>
<td>Specify the ID of the asset group(s) to report on. Multiple asset group IDs are comma separated. We'll report on all the IPs in the asset groups.</td>
</tr>
<tr>
<td>tag_set_by={name</td>
<td>id}</td>
</tr>
<tr>
<td>tag_include_selector=</td>
<td>Specify ALL to match all the asset tags for the hosts you want to report on (This is an AND operation). Specifying ANY will match any of the assets tags (This is an OR operation).</td>
</tr>
<tr>
<td>tag_set_include={value}</td>
<td>Specify asset tags for the hosts you want to report on. We'll find the hosts in your account that match your tag selection and include them in the report. Multiple tags can be provided using comma separated values.</td>
</tr>
<tr>
<td>tag_exclude_selector=</td>
<td>Specify ALL to match all the asset tags for the hosts you do not want to report on (This is an AND operation). Specifying ANY will match any of the assets tags (This is an OR operation).</td>
</tr>
<tr>
<td>tag_set_exclude={value}</td>
<td>Specify asset tags for the hosts you do not want to report on. We'll find the hosts in your account that match your tag selection and exclude them from the report. Multiple tags can be provided using comma separated values.</td>
</tr>
<tr>
<td>network={value}</td>
<td>(Valid only when the Networks feature is enabled for your account.) A network name containing the IPs to include. For a new template the default network is Global Default Network.</td>
</tr>
<tr>
<td>ips={value}</td>
<td>IP addresses to include in the report. Multiple IPs are comma separated.</td>
</tr>
</tbody>
</table>
| group_by=[HOST|PATCH|OS|AG] | Sort and group the results of the report by any of the following:  
|                            | Host = HOST 
|                            | Patch = PATCH 
|                            | Operating System = OS 
|                            | Asset Group = AG  

When include_cloud_metadata=1 is specified, then only group_by=HOST is supported.
### VM Report Templates

**Patch Template**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>include_cloud_metadata=</td>
<td>(Optional) Specify 1 to include cloud metadata for your cloud assets. Only cloud metadata for AWS is supported at this time. When not specified during a create request, a value of 0 is used. When not specified during an update request, the previous value saved in the template is kept.</td>
</tr>
<tr>
<td>include_table_of_qids_fixed=</td>
<td>Specify 1 to include QIDs that will be fixed by each patch.</td>
</tr>
<tr>
<td>include_patch_links=</td>
<td>Specify 1 to include the available links for each patch.</td>
</tr>
<tr>
<td>include_patches_from_unspecified_vendors=</td>
<td>Specify 1 to include patches from unspecified vendors.</td>
</tr>
<tr>
<td>patch_severity_by=</td>
<td>Specify assigned to display severity which is assigned to the QID for the patch detection. Specify highest to display the severity which is highest across all QIDs found on the host that can be patched.</td>
</tr>
<tr>
<td>patch_cvss_score_by=</td>
<td>Specify the CVSS version score you want to display in reports. assigned - CVSS score assigned to the QID for the patch detection highest - CVSS score highest across all QIDs found on the host that can be patched. none - Do not display CVSS scores.</td>
</tr>
<tr>
<td>cvss=</td>
<td>Specify the CVSS version score you want to display in reports. all - both CVSS versions cvssv2 - CVSS version 2 cvssv3 - CVSS version 3</td>
</tr>
<tr>
<td>display_custom_footer=</td>
<td>Specify 1 to include custom text in the report footer.</td>
</tr>
</tbody>
</table>

Notes for update patch template request:

- If include_cloud_metadata is set to 0 in the template, then you can change the group_by option to any supported value (HOST, PATCH, OS, AG).

- If include_cloud_metadata is set to 1 in the template and you change the group_by option to a value other than HOST during an update request, then we will automatically disable the cloud metadata option and we’ll show a notification in the response, letting you know that the option was disabled as a result of the change.

- If group_by is set to a value other than HOST in the template and you specify include_cloud_metadata=1 during an update request, then an error will occur because include_cloud_metadata can only have a value of 1 when group_by is set to HOST.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>display_custom_footer_text={value}</td>
<td>Specify custom text like a disclosure statement or data classification (e.g. Public, Confidential). The text you enter will appear in all reports generated from this template, except reports in XML and CSV formats. Length is maximum 4000 characters.</td>
</tr>
<tr>
<td>exclude_account_id={0</td>
<td>1}</td>
</tr>
<tr>
<td>Filters</td>
<td>Filter options such as vulnerabilities, QIDs, patches.</td>
</tr>
<tr>
<td>selective_vulns={complete</td>
<td>custom}</td>
</tr>
<tr>
<td>search_list_ids={value}</td>
<td>Specify QID to be included in the report. Multiple QIDs can be provided using values separated by a comma. This parameter is required only if selective_vulns=custom.</td>
</tr>
<tr>
<td>exclude_qid_option={0</td>
<td>1}</td>
</tr>
<tr>
<td>exclude_search_list_ids={value}</td>
<td>Specify QID to be excluded from the report. Multiple QIDs can be provided using values separated by a comma. This parameter is required only if exclude_qid_option=1.</td>
</tr>
<tr>
<td>display_non_running_kernels={0</td>
<td>1}</td>
</tr>
<tr>
<td>exclude_non_running_kernel={0</td>
<td>1}</td>
</tr>
<tr>
<td>exclude_non_running_services={0</td>
<td>1}</td>
</tr>
<tr>
<td>exclude_qids_not_exploitable_due_to_configuration={0</td>
<td>1}</td>
</tr>
<tr>
<td>selective_patches={complete</td>
<td>custom}</td>
</tr>
<tr>
<td>exclude_patch_qid_option={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
Sample Create Patch Template

API request:
```
```

Where patch_create.xml is an XML file that contains the patch template settings:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE REPORTTEMPLATE SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/report/template/patch/patchreporttemplate_info.dtd">
<REPORTTEMPLATE>
  <PATCHTEMPLATE>
    <TITLE>
      patch_search_list_ids={value}
      Specify patch QID to be included in the report. Multiple patch QIDs can be provided using values separated by a comma. This parameter is required only if selective_patches=custom.
    </TITLE>
    <exclude_patch_search_list_ids={value}>
      Specify patch QID to be excluded from the report. Multiple patch QIDs can be provided using values separated by a comma. This parameter is required only if exclude_patch_qid_option=1.
    </exclude_patch_search_list_ids>
    <found_since_days=[7|30|90|365|NoLimit]>
      Show only patches for vulnerabilities detected during the specified period of time in days. Specify NoLimit for no time limit.
    </found_since_days>
    <User Access>
      Control user access to template and reports generated from template.
    </User Access>
    <global=[0|1]>
      Share this report template with other users by making it global. Specify 1 to make it global.
    </global>
    <report_access_users={value}>
      Specify the username to share the report with a user who wouldn't already have access to the report. Multiple usernames can be provided using values separated by a comma. Each user you add will be able to view reports generated from this template even if they don't have access to the IPs in the report.
    </report_access_users>
  </PATCHTEMPLATE>
</REPORTTEMPLATE>
```
<INFO key="title"><![CDATA[My Patch Report]]></INFO>
<INFO key="owner"><![CDATA[225889]]></INFO>
</TITLE>
<TARGET>
<INFO key="patch_evaluation"><![CDATA[qidbased]]></INFO>
<INFO key="asset_groups"><![CDATA[AG1, AG2, AG3]]></INFO>
<INFO key="ips"><![CDATA[]]></INFO>
</TARGET>
<DISPLAY>
<INFO key="group_by"><![CDATA[HOST]]></INFO>
<INFO key="include_table_of_qids_fixed"><![CDATA[0]]></INFO>
<INFO key="include_patch_links"><![CDATA[0]]></INFO>
<INFO key="include_patches_from_unspecified_vendors"><![CDATA[0]]></INFO>
<INFO key="patch_severity_by"><![CDATA[assigned]]></INFO>
<INFO key="patch_cvss_score_by"><![CDATA[none]]></INFO>
<INFO key="cvss"><![CDATA[all]]></INFO>
<INFO key="display_custom_footer"><![CDATA[0]]></INFO>
<INFO key="display_custom_footer_text"><![CDATA[]]></INFO>
<INFO key="exclude_account_id"><![CDATA[0]]></INFO>
<INFO key="include_cloud_metadata"><![CDATA[1]]></INFO>
</DISPLAY>
<FILTER>
<INFO key="selective_vulns"><![CDATA[complete]]></INFO>
<INFO key="exclude_qid_option"><![CDATA[0]]></INFO>
<INFO key="display_non_running_kernels"><![CDATA[0]]></INFO>
<INFO key="exclude_non_running_kernel"><![CDATA[0]]></INFO>
<INFO key="exclude_non_running_services"><![CDATA[0]]></INFO>
<INFO key="exclude_qids_not_exploitable_due_to_configuration"><![CDATA[0]]></INFO>
<INFO key="selective_patches"><![CDATA[complete]]></INFO>
<INFO key="exclude_patch_qid_option"><![CDATA[0]]></INFO>
<INFO key="found_since_days"><![CDATA[30]]></INFO>
</FILTER>
<USERACCESS>
<INFO key="report_access_users"><![CDATA[]]></INFO>
<INFO key="global"><![CDATA[1]]></INFO>
</USERACCESS>
</PATCHTEMPLATE>
</REPORTTEMPLATE>
Sample Update Patch Template

**API request:**
```
```

Where patch_update.xml is an XML file that contains the patch template settings. See “Sample Create Patch Template” for more information.

**XML output (Success):**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/report/template/patch/update/output.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2021-06-18T10:39:12Z</DATETIME>
    <TEXT>Patch Report Template Successfully Updated</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>5062219</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

**XML output (Success with Notification):**
```
<?xml version="1.0" encoding="UTF-8" ?>
```
Map Template

/api/2.0/fo/report/template/map/

Perform actions such as create, update, delete and export on the Map Template.

Map Template Request
A summary of API Endpoint URLs is provided below.

<table>
<thead>
<tr>
<th>Action</th>
<th>API Endpoint /required parameters</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Map Template</td>
<td>&lt;base_url&gt;/api/2.0/fo/report/template/map/ Required parameters: action=create report_format=xml</td>
<td>POST</td>
</tr>
<tr>
<td>Update Map Template</td>
<td>&lt;base_url&gt;/api/2.0/fo/report/template/map/ Required parameters: template_id={value} action=update report_format=xml</td>
<td>PUT</td>
</tr>
<tr>
<td>Delete Map Template</td>
<td>&lt;base_url&gt;/api/2.0/fo/report/template/map/ Required parameters: template_id={value} action=delete</td>
<td>POST</td>
</tr>
<tr>
<td>Export Map Template</td>
<td>&lt;base_url&gt;/api/2.0/fo/report/template/map/ Required parameters: action=export report_format=xml Optional parameter: template_id=[value] When unspecified all templates for the report type get exported.</td>
<td>GET</td>
</tr>
</tbody>
</table>

Map Template settings
These parameters (all optional) are used for a create and update requests. When creating a new template the default value is shown in bold where applicable.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title=[value]</td>
<td>A string value for the title. Length is maximum 64 characters.</td>
</tr>
<tr>
<td>owner=[value]</td>
<td>Username of the owner of this template. Validity of the owner to create reports is based on the user role or business unit. See About template owner.</td>
</tr>
<tr>
<td>global=[0</td>
<td>1]</td>
</tr>
</tbody>
</table>
### Parameter Description

**Display**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| map_sort_by={ipaddress|dns|netbios|router|operatingsystem}          | Sort and group the results of the report by any of the following:  
|                               | IP Address = ipaddress  
|                               | DNS = dns  
|                               | NetBIOS = netbios  
|                               | Router = router  
|                               | Operating System = OS                                                     |
| map_related_info_lastscandate={0|1}                                | Specify 1 to include the last scan date.                                   |
| map_related_info_assetgroups={0|1}                                    | Specify 1 to include the asset groups.                                     |
| map_related_info_authenticationrecords={0|1}                           | Specify 1 to include the authentication records.                           |
| map_related_info_discoverymethod={0|1}                                 | Specify 1 to include the discovery method.                                 |
| display_custom_footer={0|1}                                             | Specify 1 to include custom text in the report footer.                     |
| display_custom_footer_text={value}                                     | Specify custom text like a disclosure statement or data classification (e.g. Public, Confidential). The text you enter will appear in all reports generated from this template, except reports in XML and CSV formats. Length is maximum 4000 characters. |
| mapexclude_account_id={0|1}                                             | Specify 1 to exclude the account login ID in the filename of downloaded reports. Use this option to remove the login ID from the filename. |

**Filters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mapIncluded_hosttypes_innetblock={0</td>
<td>1}</td>
</tr>
<tr>
<td>mapIncluded_hosttypes_scannable={0</td>
<td>1}</td>
</tr>
<tr>
<td>mapIncluded_hosttypes_livelive={0</td>
<td>1}</td>
</tr>
<tr>
<td>mapIncluded_hosttypes_approved={0</td>
<td>1}</td>
</tr>
<tr>
<td>mapIncluded_hosttypes_outofnetblock={0</td>
<td>1}</td>
</tr>
<tr>
<td>mapIncluded_hosttypes_notscannable={0</td>
<td>1}</td>
</tr>
<tr>
<td>mapIncluded_hosttypes_notlive={0</td>
<td>1}</td>
</tr>
<tr>
<td>mapIncluded_hosttypes_rogue={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

**Included Discovery Methods**

Specify at least one.
**VM Report Templates**

**Map Template**

Global report templates may be owned by Managers and Unit Managers. Non-global report templates may be owned by Managers, Unit Managers, Scanners and Readers.

Managers/Unit Managers can assign only those users as template owners who are part of their hierarchy and are added in their subscription.

**Parameter** | **Description**
--- | ---
map_idm_tcp={0|1} | Specify 1 to filter the report by discovery methods - TCP.
map_idm_udp={0|1} | Specify 1 to filter the report by discovery methods - UDP.
map_idm_traceroute={0|1} | Specify 1 to filter the report by discovery methods - TraceRoute.
map_idm_other={0|1} | Specify 1 to filter the report by discovery methods - Other.
map_idm_dns={0|1} | Specify 1 to filter the report by discovery methods - DNS.
map_idm_icmp={0|1} | Specify 1 to filter the report by discovery methods - ICMP.
map_idm_auth={0|1} | Specify 1 to filter the report by discovery methods - AUTH.

**Included Status Levels**

Included Status Levels Only applicable for differential map reports.

map_included_statuses_added={0|1} | Specify 1 to filter the report by statuses - Added.
map_included_statuses_removed={0|1} | Specify 1 to filter the report by statuses - Removed.
map_included_statuses_active={0|1} | Specify 1 to filter the report by statuses - Active.

dns_exclusions={none|DNS|DNS-DNSZone} | Exclude hosts discovered only via:
none = None
DNS = DNS
DNS-DNSZone = DNS and/or DNS Zone Transfer

included_os={value} | Specify the operating system name to filter hosts. For example, to only report on Linux hosts make sure you provide the operating system name for Linux. Multiple operating system names can be provided using values separated by a comma. Specify ALL to include all operating systems. See Identified OS.

**Samples**

Refer to Scan template examples for create, update, delete and export sample requests. Requests and outputs for Map template are similar.

**About template owner**

The user who created the report template is the owner by default. Managers and Unit Managers have the option to specify/change the owner while creating a
### Identified OS

Operating Systems identified by our service as of March 2017 are listed below.

Looking for a more current listing? Sure thing. Just log in to your Qualys account and go to Help > About.

Tip - In API requests replace spaces in OS names with underscores. For example, **Apple IOS** must be specified as **Apple_IOS**

<table>
<thead>
<tr>
<th>3Com</th>
<th>3Com HomeConnect</th>
</tr>
</thead>
<tbody>
<tr>
<td>3Com</td>
<td>3Com NBX</td>
</tr>
<tr>
<td>3Com</td>
<td>3Com OfficeConnect</td>
</tr>
<tr>
<td>3Com</td>
<td>3Com SuperStack</td>
</tr>
<tr>
<td>3Com</td>
<td>3Com Switch</td>
</tr>
<tr>
<td>3Com</td>
<td>3Com Wireless Access Point</td>
</tr>
<tr>
<td>AB</td>
<td>AB ControlLogix</td>
</tr>
<tr>
<td>Adic</td>
<td>Adic Scalar</td>
</tr>
<tr>
<td>Adic</td>
<td>Adic Storage</td>
</tr>
<tr>
<td>ADIC</td>
<td>ADIC Storage</td>
</tr>
<tr>
<td>Adtran</td>
<td>Adtran Device</td>
</tr>
<tr>
<td>Adtran</td>
<td>Adtran NetVanta</td>
</tr>
<tr>
<td>Adtran</td>
<td>Adtran TSUIQ</td>
</tr>
<tr>
<td>ADTX</td>
<td>ADTX ArrayMasStor</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 4.2-4.3</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 4.3</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 4.3.2.0-4.3.3.0</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 4.33</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 4.3-5.1</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 4.x</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 4.x-5.x</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 5.1</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 5.1-5.2</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 5.1-5.3</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 5.2</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 5.3</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 5.3.0.4</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 5.x</td>
</tr>
<tr>
<td>AIX</td>
<td>AIX 6.x</td>
</tr>
<tr>
<td>Alcatel</td>
<td>Alcatel OmniStack</td>
</tr>
<tr>
<td>Alcatel</td>
<td>Alcatel OmniSwitch</td>
</tr>
<tr>
<td>Allied</td>
<td>Allied Telesyn Switch</td>
</tr>
<tr>
<td>Alteon</td>
<td>Alteon ACE Switch</td>
</tr>
<tr>
<td>Alteon</td>
<td>Alteon Switch</td>
</tr>
<tr>
<td>Altium</td>
<td>Altium Wireless Device</td>
</tr>
<tr>
<td>Amazon</td>
<td>Amazon Linux</td>
</tr>
<tr>
<td>AMX</td>
<td>AMX Modero</td>
</tr>
<tr>
<td>APC</td>
<td>APC InfraStruXure</td>
</tr>
<tr>
<td>APC</td>
<td>APC MasterSwitch</td>
</tr>
<tr>
<td>APC</td>
<td>APC Network</td>
</tr>
<tr>
<td>APC</td>
<td>APC Network Management Card AOS</td>
</tr>
<tr>
<td>APC</td>
<td>APC Smart-UPS</td>
</tr>
<tr>
<td>AppCelera</td>
<td>AppCelera ICX</td>
</tr>
<tr>
<td>Apple</td>
<td>Apple Apple Airport Wireless Access Point</td>
</tr>
<tr>
<td>Apple</td>
<td>Apple iOS</td>
</tr>
<tr>
<td>Apple</td>
<td>Apple Wireless Access Point</td>
</tr>
<tr>
<td>Arescom</td>
<td>Arescom Device</td>
</tr>
<tr>
<td>Arescom</td>
<td>Arescom NetDSL</td>
</tr>
<tr>
<td>Ascend</td>
<td>Ascend Router</td>
</tr>
<tr>
<td>Ascent</td>
<td>Ascent Router</td>
</tr>
<tr>
<td>ASUS</td>
<td>ASUS</td>
</tr>
<tr>
<td>ASUS</td>
<td>ASUS Wireless</td>
</tr>
<tr>
<td>ASUS</td>
<td>ASUS Wireless Access Point</td>
</tr>
<tr>
<td>Aten</td>
<td>Aten KVM Switch</td>
</tr>
<tr>
<td>ATT</td>
<td>ATT NetGate</td>
</tr>
<tr>
<td>ATTO</td>
<td>ATTO Device</td>
</tr>
<tr>
<td>AudioCodes</td>
<td>AudioCodes VOIP</td>
</tr>
<tr>
<td>Avaya</td>
<td>Avaya</td>
</tr>
<tr>
<td>Avaya</td>
<td>Avaya Device</td>
</tr>
<tr>
<td>Avaya</td>
<td>Avaya G350</td>
</tr>
<tr>
<td>Avaya</td>
<td>Avaya IP Phone</td>
</tr>
<tr>
<td>Avaya</td>
<td>Avaya Wireless Access Point</td>
</tr>
<tr>
<td>Avocent</td>
<td>Avocent CCM Appliance</td>
</tr>
<tr>
<td>Axis</td>
<td>Axis</td>
</tr>
<tr>
<td>Axis</td>
<td>Axis Network Camera</td>
</tr>
<tr>
<td>Axis</td>
<td>Axis Printer</td>
</tr>
</tbody>
</table>
VM Report Templates
Map Template

Axis Storpoint CD
Axis Video Server
Axis Wireless Access Point
Axonix SuperCD
Bay Networks
Bay Networks Router
Bay Networks Switch
Belkin
Belkin Wireless Access Point
BeOS 5
BlueCoat Security Gateway
BlueSocket Embedded Linux 2.4-2.6
BorderWare Firewall
Brocade Device
Brother Printer
BSD
BSD Unix
BSDI BSD
BT Voyager
Buffalo Wireless Access Point
Cabletron
Cabletron SmartSTACK
Cabletron Switch
Caldera
Caldera Open Linux
Caldera Open UNIX 7
Caldera Open UNIX 8
Canon
Canon Network Printer
Canon Print Server
Canon Printer
Cayman3000
CEKAB Device
CentOS
CentOS
CheckPoint
CheckPoint FW1
CheckPoint FW1 NG
CheckPoint FW1 on Solaris
CheckPoint SecurePlatform
Cintech Switch
Cirronet Wireless Access Point
Cisco
Cisco Analog Phone Gateway
Cisco Analog Telephone Adaptor
Cisco Arrowpoint WebNS
Cisco ASA
Cisco Catalyst
Cisco Content Engine
Cisco Content Services Switch
Cisco Content Switching Solution
Cisco Content/File Engine
Cisco Controller
Cisco File Engine
Cisco Firewall Services Module
Cisco IOS
Cisco IP Phone
Cisco IP/TV Program Manager
Cisco Local Director
Cisco PIX
Cisco VPN
Cisco WGB350
Cisco Wireless Access Point
ClearPath MCP
CNT UltraNet Edge
Cognitive Printer
CometLabs Switch
Compaq
Compaq Insight Manager
Compaq Switch
Computone Device
Connect2Air Wireless Access Point
ControlLogix ENET
Crossroads Storage Router
Custom Micro Device
CyberGuard Firewall
CyberGuard Firewall
Datamax I-Class
Datamax Printer
Dawning SNI
Debian
Dell
Dell Laser
Dell PowerConnect
Dell PowerVault
Dell Remote Access Controller
Digi
Digi One PortServer
Digi One SP
Digi Port Server
Divar Video Camera
D-Link
D-Link DSL Modem
D-Link Print Server
D-Link Router
D-Link Switch
D-Link Wireless Access Point
Draytek Router
<table>
<thead>
<tr>
<th>VM Report Templates</th>
<th>Map Template</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Juniper Networks Application</strong></td>
<td><strong>Linux 3.0</strong></td>
</tr>
<tr>
<td><strong>Acceleration Platform DX</strong></td>
<td><strong>Linux Based MRV LX Series Server</strong></td>
</tr>
<tr>
<td><strong>Juniper Networks JUNOS</strong></td>
<td><strong>Linux behind</strong></td>
</tr>
<tr>
<td><strong>Kentrox</strong></td>
<td><strong>Lucent</strong></td>
</tr>
<tr>
<td><strong>Kentrox Q2200 Router</strong></td>
<td><strong>Lucent Cajun</strong></td>
</tr>
<tr>
<td><strong>Konica</strong></td>
<td><strong>Lucent MAX</strong></td>
</tr>
<tr>
<td><strong>Konica Minolta</strong></td>
<td><strong>Lucent Orinoco</strong></td>
</tr>
<tr>
<td><strong>Konica Printer</strong></td>
<td><strong>Lucent PBX</strong></td>
</tr>
<tr>
<td><strong>Kyocera</strong></td>
<td><strong>Lucent Router</strong></td>
</tr>
<tr>
<td><strong>Kyocera Mita</strong></td>
<td><strong>Lucent WAP</strong></td>
</tr>
<tr>
<td><strong>Kyocera Printer</strong></td>
<td><strong>LynxOS</strong></td>
</tr>
<tr>
<td><strong>Lancast</strong></td>
<td><strong>MacOS</strong></td>
</tr>
<tr>
<td><strong>Lancast Media Converter</strong></td>
<td><strong>MacOS 10.0.x-10.1.x</strong></td>
</tr>
<tr>
<td><strong>Lanier</strong></td>
<td><strong>MacOS 10.10</strong></td>
</tr>
<tr>
<td><strong>Lanier Printer</strong></td>
<td><strong>MacOS 10.11</strong></td>
</tr>
<tr>
<td><strong>Lantronix</strong></td>
<td><strong>MacOS 10.12</strong></td>
</tr>
<tr>
<td><strong>Lantronix CoBox</strong></td>
<td><strong>MacOS 10.3-10.4</strong></td>
</tr>
<tr>
<td><strong>Lantronix ET532PR</strong></td>
<td><strong>MacOS 8</strong></td>
</tr>
<tr>
<td><strong>Lantronix MSS100</strong></td>
<td><strong>MacOS 9</strong></td>
</tr>
<tr>
<td><strong>Lantronix Printer</strong></td>
<td><strong>MacOS X</strong></td>
</tr>
<tr>
<td><strong>Leitch</strong></td>
<td><strong>magicolor</strong></td>
</tr>
<tr>
<td><strong>Lexmark</strong></td>
<td><strong>magicolor 2300 Printer</strong></td>
</tr>
<tr>
<td><strong>Lexmark Optra</strong></td>
<td><strong>magicolor 3300 Printer</strong></td>
</tr>
<tr>
<td><strong>Lexmark Print Server</strong></td>
<td><strong>magicolor Printer</strong></td>
</tr>
<tr>
<td><strong>Lexmark Printer</strong></td>
<td><strong>MarkNet Pro Printer</strong></td>
</tr>
<tr>
<td><strong>LinkCom</strong></td>
<td><strong>Meditech MAGIC</strong></td>
</tr>
<tr>
<td><strong>LinkCom Xpress Print Server</strong></td>
<td><strong>MGE Uninterruptible Power Supply Systems</strong></td>
</tr>
<tr>
<td><strong>Linksys</strong></td>
<td><strong>Microtest DiscZerver</strong></td>
</tr>
<tr>
<td><strong>Linksys Router</strong></td>
<td><strong>MiLAN</strong></td>
</tr>
<tr>
<td><strong>Linksys Wireless</strong></td>
<td><strong>MiLAN Print Server</strong></td>
</tr>
<tr>
<td><strong>Linux</strong></td>
<td><strong>MiLAN Switch</strong></td>
</tr>
<tr>
<td><strong>Linux 1.2.8-1.2.13</strong></td>
<td><strong>MiraPoint</strong></td>
</tr>
<tr>
<td><strong>Linux 2.0</strong></td>
<td><strong>Mitel PBX</strong></td>
</tr>
<tr>
<td><strong>Linux 2.0.29</strong></td>
<td><strong>Motorola HomeNet WR850G</strong></td>
</tr>
<tr>
<td><strong>Linux 2.0.34-38</strong></td>
<td><strong>Moxa</strong></td>
</tr>
<tr>
<td><strong>Linux 2.1.19-2.2.20</strong></td>
<td><strong>Moxa Async Server</strong></td>
</tr>
<tr>
<td><strong>Linux 2.2</strong></td>
<td><strong>Moxa NPort Serial Server</strong></td>
</tr>
<tr>
<td><strong>Linux 2.2.20</strong></td>
<td><strong>Multi-Tech</strong></td>
</tr>
<tr>
<td><strong>Linux 2.4</strong></td>
<td><strong>Multi-Tech CommPlete</strong></td>
</tr>
<tr>
<td><strong>Linux 2.4.0-2.5.20</strong></td>
<td><strong>Multi-Tech MultiVOIP</strong></td>
</tr>
<tr>
<td><strong>Linux 2.4.20-2.4.25</strong></td>
<td><strong>Muratec MFX Printer</strong></td>
</tr>
<tr>
<td><strong>Linux 2.4.20-3</strong></td>
<td><strong>NCR Unix</strong></td>
</tr>
<tr>
<td><strong>Linux 2.4.22</strong></td>
<td><strong>NEC Projector</strong></td>
</tr>
<tr>
<td><strong>Linux 2.4.7</strong></td>
<td><strong>Neoteris Instant Virtual Extranet</strong></td>
</tr>
<tr>
<td><strong>Linux 2.4.x</strong></td>
<td><strong>NetApp</strong></td>
</tr>
<tr>
<td><strong>Linux 2.4-2.6</strong></td>
<td><strong>NetApp behind FW1</strong></td>
</tr>
<tr>
<td><strong>Linux 2.6</strong></td>
<td><strong>NetBlazer</strong></td>
</tr>
<tr>
<td><strong>Linux 2.x</strong></td>
<td><strong>NetBSD</strong></td>
</tr>
<tr>
<td>Device Name</td>
<td>Manufacturer</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>NETBuilder Bridge</td>
<td>Netgear</td>
</tr>
<tr>
<td>Netgear GSM</td>
<td></td>
</tr>
<tr>
<td>Netgear Print Server</td>
<td></td>
</tr>
<tr>
<td>Netgear Printer</td>
<td></td>
</tr>
<tr>
<td>Netgear Router</td>
<td></td>
</tr>
<tr>
<td>Netgear Smart Switch</td>
<td></td>
</tr>
<tr>
<td>Netgear Switch</td>
<td></td>
</tr>
<tr>
<td>Netgear Wireless Access Point</td>
<td></td>
</tr>
<tr>
<td>Netopia</td>
<td>Netgear</td>
</tr>
<tr>
<td>Netopia Router</td>
<td></td>
</tr>
<tr>
<td>Netphone</td>
<td></td>
</tr>
<tr>
<td>Netphone IP Phone</td>
<td></td>
</tr>
<tr>
<td>NetScaler</td>
<td></td>
</tr>
<tr>
<td>NetScaler VPN Device</td>
<td></td>
</tr>
<tr>
<td>NetScreen</td>
<td></td>
</tr>
<tr>
<td>NetScreen 100</td>
<td></td>
</tr>
<tr>
<td>NetScreen 50</td>
<td></td>
</tr>
<tr>
<td>NetScreen 5XP</td>
<td></td>
</tr>
<tr>
<td>NetSilicon Device</td>
<td></td>
</tr>
<tr>
<td>Netsilicon Device</td>
<td></td>
</tr>
<tr>
<td>NetWare</td>
<td></td>
</tr>
<tr>
<td>NetWare 4.11-5.0 SP5</td>
<td></td>
</tr>
<tr>
<td>NetWare 5</td>
<td></td>
</tr>
<tr>
<td>NetWare 5.0</td>
<td></td>
</tr>
<tr>
<td>NetWare 5.1</td>
<td></td>
</tr>
<tr>
<td>NetWare 6</td>
<td></td>
</tr>
<tr>
<td>NetWare 6.5</td>
<td></td>
</tr>
<tr>
<td>NetWare Print Server</td>
<td></td>
</tr>
<tr>
<td>Network Camera</td>
<td></td>
</tr>
<tr>
<td>Network Print Server</td>
<td></td>
</tr>
<tr>
<td>Network Printer</td>
<td></td>
</tr>
<tr>
<td>Network Scanner</td>
<td></td>
</tr>
<tr>
<td>NGS 500 Router</td>
<td></td>
</tr>
<tr>
<td>NIB Network Printer</td>
<td></td>
</tr>
<tr>
<td>Nokia</td>
<td></td>
</tr>
<tr>
<td>Nokia IPSO</td>
<td></td>
</tr>
<tr>
<td>Nokia Wireless Access Point</td>
<td></td>
</tr>
<tr>
<td>Nortel</td>
<td></td>
</tr>
<tr>
<td>Nortel Device</td>
<td></td>
</tr>
<tr>
<td>Nortel Networks BayStack</td>
<td></td>
</tr>
<tr>
<td>Nortel Passport</td>
<td></td>
</tr>
<tr>
<td>Nortel Router</td>
<td></td>
</tr>
<tr>
<td>Nortel Switch</td>
<td></td>
</tr>
<tr>
<td>NRG</td>
<td></td>
</tr>
<tr>
<td>NRG Network</td>
<td></td>
</tr>
<tr>
<td>NRG Printer</td>
<td></td>
</tr>
<tr>
<td>Okidata Printer</td>
<td></td>
</tr>
<tr>
<td>OkiLAN Print Server</td>
<td></td>
</tr>
<tr>
<td>Open Networks Router</td>
<td></td>
</tr>
<tr>
<td>OpenBSD</td>
<td></td>
</tr>
<tr>
<td>Oracle Enterprise Linux</td>
<td></td>
</tr>
<tr>
<td>Oracle Enterprise Linux 4.5</td>
<td></td>
</tr>
<tr>
<td>Oracle Enterprise Linux 5.2</td>
<td></td>
</tr>
<tr>
<td>Orinoco Wireless Access Point</td>
<td></td>
</tr>
<tr>
<td>Packeteer</td>
<td></td>
</tr>
<tr>
<td>Packeteer PacketSeeker</td>
<td></td>
</tr>
<tr>
<td>Packeteer PacketShaper</td>
<td></td>
</tr>
<tr>
<td>Panasonic Network Camera</td>
<td></td>
</tr>
<tr>
<td>Paradyne Device</td>
<td></td>
</tr>
<tr>
<td>Perle Jetstream</td>
<td></td>
</tr>
<tr>
<td>PocketPro Print Server</td>
<td></td>
</tr>
<tr>
<td>Point Six Point Server</td>
<td></td>
</tr>
<tr>
<td>Polycom</td>
<td></td>
</tr>
<tr>
<td>Polycom Device</td>
<td></td>
</tr>
<tr>
<td>Polycom MGC</td>
<td></td>
</tr>
<tr>
<td>Polycom VSX</td>
<td></td>
</tr>
<tr>
<td>Power Measurement ION Meter</td>
<td></td>
</tr>
<tr>
<td>Powerware</td>
<td></td>
</tr>
<tr>
<td>Powerware ConnectUPS</td>
<td></td>
</tr>
<tr>
<td>Powerware UPS Device</td>
<td></td>
</tr>
<tr>
<td>Precidia Device</td>
<td></td>
</tr>
<tr>
<td>Primergy RSB</td>
<td></td>
</tr>
<tr>
<td>Printronix Printer</td>
<td></td>
</tr>
<tr>
<td>Procom NetFORCE</td>
<td></td>
</tr>
<tr>
<td>pSOSSystem</td>
<td></td>
</tr>
<tr>
<td>QNX</td>
<td></td>
</tr>
<tr>
<td>Quantum</td>
<td></td>
</tr>
<tr>
<td>Quantum NAS SnapServer</td>
<td></td>
</tr>
<tr>
<td>Quantum PX506 Tape Library</td>
<td></td>
</tr>
<tr>
<td>Quick Eagle Device</td>
<td></td>
</tr>
<tr>
<td>RadiSys iRMX</td>
<td></td>
</tr>
<tr>
<td>Radware Device</td>
<td></td>
</tr>
<tr>
<td>Raptor Firewall</td>
<td></td>
</tr>
<tr>
<td>Red Hat</td>
<td></td>
</tr>
<tr>
<td>Redline</td>
<td></td>
</tr>
<tr>
<td>Redline Networks Processor</td>
<td></td>
</tr>
<tr>
<td>Redline Wireless Access Point</td>
<td></td>
</tr>
<tr>
<td>Ricoh</td>
<td></td>
</tr>
<tr>
<td>RICOH Aficio</td>
<td></td>
</tr>
<tr>
<td>Ricoh Aficio</td>
<td></td>
</tr>
<tr>
<td>Ricoh Printer</td>
<td></td>
</tr>
<tr>
<td>Ringdale Device</td>
<td></td>
</tr>
<tr>
<td>RIO Xtreme</td>
<td></td>
</tr>
<tr>
<td>RiverStone Networks Router</td>
<td></td>
</tr>
<tr>
<td>RoamAbout R2</td>
<td></td>
</tr>
<tr>
<td>Rockwell</td>
<td></td>
</tr>
<tr>
<td>VM Report Templates</td>
<td>Map Template</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Rockwell Automation</td>
<td>Solaris 8-10</td>
</tr>
<tr>
<td>S3Wireless Wireless Access Point</td>
<td>Solaris 9</td>
</tr>
<tr>
<td>Savin Printer</td>
<td>Solaris 9-10</td>
</tr>
<tr>
<td>Scannex NetBuffer</td>
<td>Solaris behind</td>
</tr>
<tr>
<td>Schneider Electric Controller</td>
<td>Spectrum24 Wireless Access Point</td>
</tr>
<tr>
<td>SCO</td>
<td>Stallion EasyServer</td>
</tr>
<tr>
<td>SCO OpenServer</td>
<td>StarDot NetCam</td>
</tr>
<tr>
<td>SCO Unix</td>
<td>Summit Switch</td>
</tr>
<tr>
<td>SCO UnixWare</td>
<td>Sun</td>
</tr>
<tr>
<td>SCO UnixWare Firewall</td>
<td>Sun Cobalt Linux</td>
</tr>
<tr>
<td>SensaTronics Environmental Monitor</td>
<td>Sun Lights Out</td>
</tr>
<tr>
<td>Sentry Remote Power Manager</td>
<td>SUN StorEdge RAID</td>
</tr>
<tr>
<td>Shark supercomputer</td>
<td>SuperScript Printer</td>
</tr>
<tr>
<td>Sharp Printer</td>
<td>SuSE</td>
</tr>
<tr>
<td>Shore Microsystems Link Protector</td>
<td>SuSE Linux 10</td>
</tr>
<tr>
<td>Sidewinder G2</td>
<td>SuSE Linux 11</td>
</tr>
<tr>
<td>Siemens</td>
<td>SuSE Linux 7</td>
</tr>
<tr>
<td>Siemens 5940 Router</td>
<td>SuSE Linux 8</td>
</tr>
<tr>
<td>Siemens HiPath 3000</td>
<td>SuSE Linux 9</td>
</tr>
<tr>
<td>Siemens I-Gate</td>
<td>Sveasoft Firmware</td>
</tr>
<tr>
<td>Siemens IP Phone</td>
<td>Symantec Raptor Firewall</td>
</tr>
<tr>
<td>Siemens Wireless Access Point</td>
<td>Symbol Wireless Access Point</td>
</tr>
<tr>
<td>Signature System</td>
<td>Symon NetLite</td>
</tr>
<tr>
<td>Silex Pricom Print Server</td>
<td>SYSTEC CAN-Ethernet Gateway</td>
</tr>
<tr>
<td>SIMATIC NET CP</td>
<td>Tandberg</td>
</tr>
<tr>
<td>SMC</td>
<td>Tandberg Device</td>
</tr>
<tr>
<td>SMC Networks SMC8624T</td>
<td>Tandem</td>
</tr>
<tr>
<td>SMC Router</td>
<td>Tandem NSK</td>
</tr>
<tr>
<td>SMC Wireless Access Point</td>
<td>Tektronix Phaser Printer</td>
</tr>
<tr>
<td>SMC2671 Wireless Access Point</td>
<td>Telindus Router</td>
</tr>
<tr>
<td>SNAP Ethernet Brain</td>
<td>Tenor Switch</td>
</tr>
<tr>
<td>Snap Server</td>
<td>TINI</td>
</tr>
<tr>
<td>Solaris</td>
<td>TiVo</td>
</tr>
<tr>
<td>Solaris 10</td>
<td>TiVo Series</td>
</tr>
<tr>
<td>Solaris 11</td>
<td>TopLayer Appsafe</td>
</tr>
<tr>
<td>Solaris 2</td>
<td>Toshiba NWcamera</td>
</tr>
<tr>
<td>Solaris 2.5.1</td>
<td>Transition Networks Device</td>
</tr>
<tr>
<td>Solaris 2.5-2.5.1</td>
<td>Trendnet Print Server</td>
</tr>
<tr>
<td>Solaris 26</td>
<td>Trendware Print Server</td>
</tr>
<tr>
<td>Solaris 2.6</td>
<td>Tru64</td>
</tr>
<tr>
<td>Solaris 2.6-10</td>
<td>Tru64 Unix 4.0d</td>
</tr>
<tr>
<td>Solaris 2.6-7</td>
<td>Tru64 Unix 5.x</td>
</tr>
<tr>
<td>Solaris 2.6-8</td>
<td>Tut Modem</td>
</tr>
<tr>
<td>Solaris 2.7</td>
<td>TV Program Manager</td>
</tr>
<tr>
<td>Solaris 5</td>
<td>U.S. Robotics</td>
</tr>
<tr>
<td>Solaris 5.8</td>
<td>U.S. Robotics Access point</td>
</tr>
<tr>
<td>Solaris 6-8</td>
<td>U.S. Robotics ADSL Wireless Gateway</td>
</tr>
<tr>
<td>Solaris 7</td>
<td>U.S. Robotics Broadband Router</td>
</tr>
<tr>
<td>Solaris 7-10</td>
<td>U.S. Robotics Wireless Access Point</td>
</tr>
<tr>
<td>Solaris 8</td>
<td></td>
</tr>
</tbody>
</table>
Identified Services

Services identified by our service as of March 2017 are listed below.

Looking for a more current listing? Just log in to your Qualys account and go to Help > About.

Tip - In API requests replace spaces in service names with underscores. For example, Blackberry Attachment must be specified as Blackberry_Attachment

<table>
<thead>
<tr>
<th>Identified Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveSync</td>
</tr>
<tr>
<td>ADDP</td>
</tr>
<tr>
<td>afpovertcp</td>
</tr>
<tr>
<td>akak_trojan</td>
</tr>
<tr>
<td>amandaidx</td>
</tr>
<tr>
<td>aml</td>
</tr>
<tr>
<td>Apple_Airport_Management</td>
</tr>
<tr>
<td>Applix</td>
</tr>
<tr>
<td>Applix_axnet</td>
</tr>
<tr>
<td>Applix_TM1_Admin_Server</td>
</tr>
<tr>
<td>Applix_TM1_Server</td>
</tr>
<tr>
<td>Arkeiad_Network_Backup</td>
</tr>
<tr>
<td>ARUGIZER_BACKDOOR</td>
</tr>
<tr>
<td>auth</td>
</tr>
</tbody>
</table>
Berlios_Global_Positioning_System_Daemon
BIGFIX_ENTERPRISE_SERVER
BITCOIN
bitkeeper
Blackberry_Attachment
BMC_Patrol
BO2K_backdoor
bofra_worm
bpcd
bpjava_msvc
cabrightstor
CA_License_Management_Agent
CA_Unicenter_Services
CENTUM_CS_3000
chargen
chargen_udp
CHECKPOINT_FW-1_CLIENT_AUTH_SERVER
chindi
cisco_cnr
CISCO_CNR_AICSERVAGT
Cisco_Secure_ACS
cisco_ta
citadel
Citrix_CMC
Citrix_ICA
CoDeSys
Cognos_Powerplay_Enterprise_Server
Computer_Associates_License_Manager
COREid_Access_Server
crystal_info
Crystal_Reports_App_Server
Crystal_Reports_CMS
cvspserver
daap
dameware
darxite
daytime
daytime_udp
DC_Directory_Server
dcerpc
dchub
DHCP_or_Bootp_Server
DNS_Server
dtspcd
echo
echo_udp
edonkey_server
EMC_EmailXtender
finger
Forte_for_Java
ftp
FW1
FW1_NG_Services
gamsoft_telsrv
GCS_SysID
GIOP
girlfriend
gnutella
gopher
h323
healthd
HoneyD_HoneyPot
HP_DATAPROTECT
HP_printer_service
hparray
hpov_alarm
HPOV_BBC
HPOV_CODA
hpov_topmd
hpov_trcsvc
http
http_over_ssl
IBM_SolidDB
IBM_DB2_Universal_Database
IBM_TIVOLI_STORAGE_MANAGER
icecast
ident
imap
INDUSOFT
Infopulse_Gatekeeper
ipmi
ipp
irc
ISA_Proxy
isakmp
ISAKMP_over_TCP
iSCSI
iSNS
jabber
Kadmin-4
kazaa
Kerberos-5
l2tp
LANDesk
LANDesk_CBA_PDS
LANDesk_MANAGEMENT_AGENT
LANDesk_MANAGEMENT_AGENT
ldap
ldap_over_ssl
limewire
linuxconf
lpd
managesoft
McAfee_ePolicy_Orchestrator
melange_chat
MERCUR_Control-Service
Micromuse_Netcool_Object_Server
microsoft-ds
Microsoft_Message_Queue_Server
minisql
modbus
MODBUS_UDP
mqseries
msdtc
MSMQ_Ping
msrpc
msrpc-over-http
msrpc_udp
msql
msql_monitor
MYDESKTOP
mysql
named_udp
ncp
nessus
netbios_ns
netbios_ssn
netbus
netop
netstat
Netviewer_PC_Duo
nfs
nntp
ntp
ocsp
ocssd
Omniquad_Server
open_vpn
openvpn
oracle
Oracle_Express_Server
Oracle_Express_Server_xsagent
Oracle_Express_Server_xsgdaemon
oracle_intelligent_agent
ORACLE_RMI
pcanywhere
pen
Polycom_MGC_Management
pop2
pop3
PostgreSQL
ppta
PRORAT_TROJAN
proxy_http
proxy_telnet
psmon
psmon
psmon
Quote_of_the_Day
quote_of_the_day_udp
radius
radius_tcp
radmin
rccmd
RealMedia_EncoderServer
Red_Carpet_Daemon
RELIABLE_DATAGRAM_SOCKETS_OVER_TCP
Resonate_CD_Agent
resource_monitor_api
Resource_Monitoring_and_Control
rip
rlogin
RMIRegistry
rpc
rpc_udp
RSA_Auth_Mgr
rsh/rexec
rsyncd
rtsp
SAP_MAXDB
SAP_Protocol
SAPgui
SGI_Performance_Copilot
shell
SHOUTcast
skinny
skype
slapper
SMS
smtp
smtp
smux
snmp
snmp2
socks4
socks5
SPLASHTOP_REMOTE_DESKTOP
VM Report Templates
Map Template

spychat
Spytech_SpyAnywhere
ssdp
ssh
ssl_over_ssl
swagentd
swat
sybase_adaptive_server
Symantec_EMS_client_server
Symantec_AntiVirus
Symantec_AntiVirus_Rtvscan
Symantec_AntiVirus_Rtvscan_UDP
SysGalUR
systat
talk
telnet
telnet_over_ssl
tftp
time
time_udp
timestamp_over_http
trendmicro_officescan
trojan_fireby
unknown
unknown_over_ssl
UPNP
ut_game_queryport
uucp
VMware_Authentication_Daemon
vnc
vnetd
vold
Volume_Manager_Storage_Administrator
VXWORKS_WDBRPC_UDP
watchguard_admin
webshield
win_remote_desktop
winmx
WINS_Replication
Wonderware_InTouch
wmserver
WSUS_SERVER
x11
X11_Font_Service
xdmcp
xinetd
Xitami
xpilot
XYZFind
Yahoo_Instant_Messenger
yeemp
ZLink

Categories
Vulnerability Categories as defined by our service as of March 2017 are listed below.

Want a current listing? No problem. Just log in to your Qualys account, go to the KnowledgeBase, click the Search button, and open the Category menu.

Looking for category descriptions? We’ve got you covered. Log in to your Qualys account, go to Help > Online Help and search for Categories and you’ll see the article on Vulnerability Categories with all the details.

Tip - In API requests replace spaces in category names with underscores. For example, Amazon Linux must be specified as Amazon_linux

AIX
Amazon Linux
Backdoors and trojan horses
Brute Force Attack
CentOS
CGI
Cisco
Database
Debian
DNS and BIND
E-Commerce
Fedora
File Transfer Protocol
Finger
Firewall
Forensics
General remote services
Hardware
HP-UX
Information gathering
Internet Explorer
Local
Mail services
Malware
News Server
NFS
OEL
Office Application
Proxy
RedHat
RPC
Security Policy
SNMP
Solaris
SMB / NETBIOS
SUSE
TCP/IP
Ubuntu
VMware
Web Application
Web Application Firewall
Web server
Windows
X-Window
VM Remediation Tickets

List, edit and delete remediation tickets, created using the VM app, in the user’s account.

Remediation Tickets overview

Ticket Parameters

View Ticket List

Edit Tickets

Delete Tickets

View Deleted Ticket List

Get Ticket Information

Set Vulnerabilities to Ignore on Hosts

Remediation Tickets overview

Qualys provides fully secure audit trails that track vulnerability status for all detected vulnerabilities. As follow up audits occur, vulnerability status levels - new, active, fixed, and re-opened - are updated automatically and identified in trend reports, giving users access to the most up-to-date security status. Using Remediation Workflow, Qualys automatically updates vulnerability status in remediation tickets, triggering ticket updates and closure in cases where vulnerabilities are verified as fixed.

Ticket information includes

**Ticket Due Date** - Each ticket has a due date for ticket resolution. The number of days allowed for ticket resolution is set as part of the policy rule configuration. Overdue tickets are those tickets for which the due date for resolution has passed.

**Ticket state/status** - Several events trigger ticket updates as described earlier. Certain ticket updates result in changes to ticket state/status as indicated below.

- Open refers to new and reopened tickets. Tickets are reopened in these cases: 1) when the service detected vulnerabilities for tickets with state/status Resolved or Closed/Fixed, and 2) when users or the service reopened Closed/Ignored tickets.

- Resolved refers to tickets marked as resolved by users.

- Closed/Fixed refers to tickets with vulnerabilities verified as fixed by the service.

- Closed/Ignored refers to tickets ignored by users or the service (based on a user policy). Also, users can ignore vulnerabilities on hosts. If tickets exist for vulnerabilities set to ignore status, the service sets them to Closed/Ignored, and if tickets do not exist for these issues the service adds new tickets and changes them to Closed/Ignored.
**Invalid tickets** - Tickets are invalid due to the changing status of the IP address or ticket owner. Regarding the IP address, a ticket is marked invalid when the ticket’s IP address is removed from the ticket owner’s account (applies to Unit Manager, Scanner, or Reader). Regarding the ticket owner, a ticket is marked invalid when the ticket owner’s account is inactive, deleted, or the user’s role was changed to Contact.
Ticket Parameters

Many ticket parameters are available for making API requests to view, update and delete active tickets and defining tickets to take actions on. Overdue and Invalid tickets are selected automatically, unless otherwise requested.

- All ticket parameters are optional and valid for these requests: ticket_list.php, ticket_edit.php and ticket_delete.php.
- At least one parameter is required.
- Multiple parameters are combined with a logical “and”.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ticket_numbers= {nnn,nnn-nnn,...}</td>
<td>Tickets with certain ticket numbers. Specify one or more ticket numbers and/or ranges. Use a dash (-) to separate the ticket range start and end. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>since_ticket_number= {value}</td>
<td>Tickets since a certain ticket number. Specify the lowest ticket number to be selected. Selected tickets will have numbers greater than or equal to the ticket number specified.</td>
</tr>
<tr>
<td>until_ticket_number= {value}</td>
<td>Tickets until a certain ticket number. Specify the highest ticket number to be selected. Selected tickets will have numbers less than or equal to the ticket number specified.</td>
</tr>
<tr>
<td>show_vuln_details={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

Ticket Properties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ticket_assignee={value}</td>
<td>Tickets with a certain assignee. Specify the user login of an active user account.</td>
</tr>
<tr>
<td>overdue={0</td>
<td>1}</td>
</tr>
<tr>
<td>invalid={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
### Ticket Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>states={state}</td>
<td>Tickets with certain ticket state/status. Specify one or more state/status codes. A valid value is OPEN (for state/status Open or Open/Reopened), RESOLVED (for state Resolved), CLOSED (for state/status Closed/Fixed), or IGNORED (for state/status Closed/Ignored). Multiple entries are comma separated.</td>
</tr>
<tr>
<td></td>
<td>To select ignored vulnerabilities on hosts, specify: states=IGNORED</td>
</tr>
<tr>
<td>modified_since_datetime={value}</td>
<td>Tickets modified since a certain date/time. Specify a date (required) and time (optional) since tickets were modified. Tickets modified on or after the date/time are selected.</td>
</tr>
<tr>
<td>unmodified_since_datetime={value}</td>
<td>Tickets not modified since a certain date/time. Specify a date (required) and time (optional) since tickets were not modified. Tickets not modified on or after the date/time are selected.</td>
</tr>
<tr>
<td>ips={nnn,nnn-nnn,...}</td>
<td>Tickets on hosts with certain IP addresses. Specify one or more IP addresses and/or ranges. Multiple entries are comma separated.</td>
</tr>
<tr>
<td>asset_groups={ag1,ag2,...}</td>
<td>Tickets on hosts with IP addresses which are defined in certain asset groups. Specify the title of one or more asset groups. Multiple asset groups are comma separated. The title &quot;All&quot; may be specified to select all IP addresses in the user account.</td>
</tr>
<tr>
<td>dns_contains={value}</td>
<td>Tickets on hosts that have a NetBIOS host name which contains a certain text string. Specify a text string to be used. This string may include a maximum of 100 characters (ascii).</td>
</tr>
<tr>
<td>netbios_contains={value}</td>
<td>Tickets on hosts that have a NetBIOS host name which contains a certain text string. Specify a text string to be used. This string may include a maximum of 100 characters (ascii).</td>
</tr>
<tr>
<td>host_id={value}</td>
<td>Tickets related to a particular asset when the specific HOST_ID is provided.</td>
</tr>
</tbody>
</table>
View Ticket List

/msp/ticket_list.php

View remediation tickets and related ticket information in the user’s account.

Basic HTTP authentication is required. Session based authentication is not supported using this API.

Using an account with more than 1,000 tickets (or potentially more than 1,000 tickets), it is recommended that you write a script that makes multiple ticket_list.php requests until all tickets are retrieved.

A maximum of 1,000 tickets can be returned from a single ticket_list.php request. If this maximum is reached, the function returns a “Truncated after 1,000 records” message at the end of the XML output with the last ticket number included. Using an account with

---

**Parameter** | **Description**
--- | ---
show_host_id=[0|1] | When unspecified or show_host_id=0, the Host ID will not appear in the XML output. Specify show_host_id=1 to show the Host ID in the output.

**Vulnerability Info**

vuln_severities={1,2,3,4,5} | Tickets for vulnerabilities with certain severity levels. Specify one or more severity levels. Multiple levels are comma separated.
potential_vuln_severities={1,2,3,4,5} | Tickets for potential vulnerabilities with certain severity levels. Specify one or more severity levels. Multiple levels are comma separated.
qids={qid,qid,...} | Tickets for vulnerabilities with certain QIDs (Qualys IDs). Specify one or more QIDs. A maximum of 10 QIDs may be specified. Multiple QIDs are comma separated.
vuln_title_contains={value} | Tickets for vulnerabilities that have a title which contains a certain text string. The vulnerability title is defined in the KnowledgeBase. Specify a text string. This string may include a maximum of 100 characters (ascii).
vuln_details_contains={value} | Tickets for vulnerabilities that have vulnerability details which contain a certain text string. Vulnerability details provide descriptions for threat, impact, solution and results (scan test results, when available). Specify a text string. This string may include a maximum of 100 characters (ascii).
vendor_ref_contains={value} | Tickets for vulnerabilities that have a vendor reference which contains a certain text string. Specify a text string. This string may include a maximum of 100 characters (ascii).
more than 1,000 tickets (or potentially more than 1,000 tickets), it is recommended that you write a script that makes multiple ticket_list.php requests until all tickets have been retrieved.

Permissions - Managers can view all tickets in the subscription. Unit Managers can view tickets for IP addresses in the user’s same business unit. Scanners and Readers can view tickets for IP addresses in the user’s own account.

Input Parameters
Click here for ticket list input parameters

Samples
View Open tickets for owner:

https://qualysapi.qualys.com/msp/ticket_list.php?
ticket_assignee=comp_ja&states=OPEN

View ticket number range:

https://qualysapi.qualys.com/msp/ticket_list.php?
ticket_numbers=001800-002800

View tickets with severity 5 confirmed vulnerabilities:

https://qualysapi.qualys.com/msp/ticket_list.php?
vuln_severities=5

View tickets that have been marked as Closed/Fixed or Closed/Ignored since June 1, 2018:


List all ignored vulnerabilities in the user’s account:

https://qualysapi.qualys.com/msp/ticket_list.php?asset_groups=All&states=IGNORED

View tickets related to SSH vulnerabilities:

https://qualysapi.qualys.com/msp/ticket_list.php?
vuln_title_contains=SSH&vuln_details contiene=SSH

View Invalid tickets for hosts in the “Desktops” or “Servers” asset groups:


View all tickets filtered by Host ID and Show Host ID:

https://qualysapi.qualys.com/msp/ticket_list.php?host_id=355311&show_host_id=1
View Overdue tickets assigned to James Adrian (comp_ja) that have not been modified since May 30, 2018 at 16:30:00 (UTC/GMT) for vulnerabilities with a severity level of 3, 4 or 5 and to include vulnerability details in the results:

```
https://qualysapi.qualys.com/msp/ticket_list.php?
unmodified_since_datetime=2018-05-30T16:30:00Z
&vuln_severities=3,4,5&overdue=1&ticket_assignee=comp_ja
&show_vuln_details=1
```

**DTD**

`<platform API server>/ticket_list_output.dtd`

**Edit Tickets**

`/msp/ticket_edit.php`

Edit remediation tickets in the user’s account. Multiple tickets can be edited at one time in bulk. Many ticket parameters are supported for selecting what tickets you’d like to edit.

Basic HTTP authentication is required. Session based authentication is not supported using this API.

Editing tickets can be a time intensive task, especially when batch editing many tickets. To ensure best performance, a maximum of 20,000 tickets can be edited in one ticket_edit.php request. It’s recommended best practice that you choose to schedule batch updates to occur when ticket processing will least impact user productivity. If the ticket_edit.php request identifies more than 20,000 tickets to be edited, then an error is returned.

Permissions - Managers can edit all tickets in the subscription. Unit Managers can edit tickets for IP addresses in the user’s same business unit. Scanners and Readers do not have permissions to edit tickets.

**Input Parameters**

Click here to view ticket parameters for selecting tickets to edit
The following parameters are used to define the ticket data to be edited. At least one of the following edit parameters is required.

**Parameter**  | **Description**  
---|---
change_assignee= {value}  | (Optional) Used to change the ticket assignee, specified by user login, in all selected tickets. The assignee’s account must have a user role other than Contact, and the hosts associated with the selected tickets must be in the user account.  
change_state= {value}  | (Optional) Used to change the ticket state/status to the specified state/status in all selected tickets. A valid value is OPEN (for state/status Open and Open/Reopened), RESOLVED (for state Resolved), or IGNORED (for state/status Closed/Ignored). See “Ticket State/Status Transitions” below for information on valid changes.  
add_comment= {value}  | (Optional) Used to add a comment in all selected tickets. The comment text may include a maximum of 2,000 characters (ascii).  
reopen_ignored_days= {value}  | (Optional) Used to reopen Closed/Ignored tickets in a set number of days. Specify the due date in N days, where N is a number of days from today. A valid value is an integer from 1 to 730.  

When the due date is reached, the ticket state is changed from Closed/Ignored to Open, assuming the issue still exists, and the ticket is marked as overdue. If the issue was resolved at some point while the ticket was in the Closed/Ignored state, then the ticket state is changed from Closed/Ignored to Closed/Fixed.

**Ticket State/Status Transitions**

The Qualys remediation workflow feature is a closed loop ticketing system for remediation management and policy compliance. Users may edit tickets to make certain ticket state changes as shown below.

<table>
<thead>
<tr>
<th>From State/Status</th>
<th>To State/Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Open</td>
</tr>
<tr>
<td>Resolved</td>
<td>valid</td>
</tr>
<tr>
<td>Closed/Ignored</td>
<td>valid</td>
</tr>
<tr>
<td>Closed/Fixed</td>
<td>valid</td>
</tr>
</tbody>
</table>

**Samples**

Edit ticket and add comment:

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X POST -d "ticket_numbers=23456&add_comment=Host+patched,+ready+for+re-scan"
```
"https://qualysapi.qualys.com/msp/ticket_edit.php?"

Edit multiple tickets to change the ticket owner to Alice Cook (acme_ac) for tickets since ticket number #00215555 (tickets with numbers greater than or equal to #00215555) which are marked invalid:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X POST -d "since_ticket_number=00215555&invalid=1&change_assignee=acme_ac" "https://qualysapi.qualys.com/msp/ticket_edit.php?"
```

Edit Open tickets on IP addresses in asset groups “New York” and “London” and change the ticket state to Ignored:

```
```

Edit Open tickets unmodified since August 1, 2017 that are assigned to Tim Burke (acme_tb) and change the ticket assignee to Alice Cook (acme_ac):

```
```

Reopen all Closed/Ignored tickets on host 10.10.10.120 in 7 days:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X POST -d "ips=10.10.10.120&reopen_ignored_days=7" "https://qualysapi.qualys.com/msp/ticket_edit.php?"
```

**DTD**

```
<platform API server>/ticket_edit_output.dtd
```

**Delete Tickets**

```
/msp.ticket_delete.php
```

Delete remediation tickets in the user’s account. Multiple tickets can be deleted at one time in bulk. Many ticket parameters are supported for selecting what tickets you’d like to edit.

Basic HTTP authentication is required. Session based authentication is not supported using this API.
Deleting tickets can be a time intensive task, especially when batch deleting many tickets. To ensure best performance, a maximum of 20,000 tickets can be deleted in one ticket_delete.php request. It’s recommended best practice that you choose to schedule batch updates to occur when ticket processing will least impact user productivity. If the ticket_delete.php request identifies more than 20,000 tickets to be deleted, then an error is returned.

Permissions - Managers can delete all tickets in the subscription. Unit Managers can delete tickets for IP addresses in their same business unit. Scanners and Readers have no permissions to delete tickets.

**Input Parameters**

Click here to view ticket parameters for selecting tickets to delete

**Samples**

Delete certain ticket number:

https://qualysapi.qualys.com/msp/ticket_delete.php?
ticket_numbers=2487

Delete tickets between ticket #001000 and ticket #002500:

https://qualysapi.qualys.com/msp/ticket_delete.php?
since_ticket_number=1000&until_ticket_number=2500

Delete Closed/Fixed tickets owned by James Adrian (comp_ja):

https://qualysapi.qualys.com/msp/ticket_delete.php?
states=CLOSED&ticket_assignee=comp_ja

Delete tickets on vulnerabilities with an assigned severity level of 1 and potential vulnerabilities with an assigned severity level of 1-3:

https://qualysapi.qualys.com/msp/ticket_delete.php?
vuln_severities=1&potential_vuln_severities=1,2,3

Delete Overdue tickets assigned to James Adrian (comp_ja) that have not been modified since July 01, 2018 at 12:00:00 (UTC/GMT)

https://qualysapi.qualys.com/msp/ticket_delete.php?
unmodified_since_datetime=2018-07-01T12:00:00Z
&overdue=1&ticket_assignee=comp_ja

**DTD**

<platform API server>/ticket_delete_output.dtd
View Deleted Ticket List

/msp/ticket_list_deleted.php

View deleted tickets in the user's account. This function may be run by Managers. The functionality provided allows for real-time integration with third-party applications.

Basic HTTP authentication is required. Session based authentication is not supported using this API.

The XML results returned by the ticket_list_deleted.php function identifies deleted tickets by ticket number and deletion date/time.

A maximum of 1,000 deleted tickets can be returned from a single ticket_list_deleted.php request. If this maximum is reached, the function returns a “Truncated after 1,000 records” message at the end of the XML report with the last ticket number included.

Permissions - Manager user role is required.

Input Parameters

All parameters are optional. At least one parameter is required. Multiple parameters are combined with a logical “and”.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ticket_numbers=</td>
<td>(Optional) Specifies certain ticket numbers. Specify one or more ticket numbers and/or ranges. Ticket range start and end is separated by a dash (-). Multiple entries are comma separated.</td>
</tr>
<tr>
<td>since_ticket_number=</td>
<td>(Optional) Specifies tickets since a certain ticket number. Specify the lowest ticket number to be selected. Selected tickets will have numbers greater than or equal to the ticket number specified.</td>
</tr>
<tr>
<td>until_ticket_number=</td>
<td>(Optional) Specifies tickets until a certain ticket number. Specify the highest ticket number to be selected. Selected tickets will have numbers less than or equal to the ticket number specified.</td>
</tr>
</tbody>
</table>
Get Ticket Information

/msp/get_tickets.php

Samples

View tickets deleted in ticket number range:

https://qualysapi.qualys.com/msp/ticket_list_deleted.php?
ticket_numbers=120-200

View tickets deleted since ticket number:

https://qualysapi.qualys.com/msp/ticket_list_deleted.php?
since_ticket_number=400

View tickets deleted since date:

https://qualysapi.qualys.com/msp/ticket_list_deleted.php?
deleted_since_datetime=2018-01-01

Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deleted_since_datetime=&lt;value&gt;</td>
<td>(Optional) Specifies tickets deleted since a certain date/time. Specify a</td>
</tr>
<tr>
<td></td>
<td>date/time. Specify a date (required) and time (optional) to identify this</td>
</tr>
<tr>
<td></td>
<td>timeframe. Tickets deleted on or after the date/time are selected.</td>
</tr>
<tr>
<td></td>
<td>date/time is specified in YYYY-MM-DD[THH:MM:SSZ]</td>
</tr>
<tr>
<td></td>
<td>format (UTC/GMT) like “2006-01-01” or “2006-05-25T23:12:00Z”.</td>
</tr>
</tbody>
</table>

| deleted_before_datetime=<value>    | (Optional) Specifies tickets deleted before a certain date/time. Specify a |
|                                    | date/time. Specify a date (required) and time (optional) to identify this  |
|                                    | timeframe. Tickets deleted on or before the date/time are selected.        |
|                                    | date/time is specified in YYYY-MM-DD[THH:MM:SSZ]                          |
|                                    | format (UTC/GMT) like “2006-01-01” or “2006-05-25T23:12:00Z”.             |

DTD

<platform API server>/ticket_list_deleted_output.dtd

Get Ticket Information

/msp/get_tickets.php

View remediation ticket information from the user’s account that can be integrated with third-party applications. Only remediation tickets that the user has permission to view are returned in the resulting ticket information report.

Basic HTTP authentication is required. Session based authentication is not supported using this API.
VM Remediation Tickets
Get Ticket Information

Qualys recommends that you run the get_tickets.php function two times a day, so that ticket updates due to the latest scan results and user productivity are made available in the ticket information reports.

Permissions - Managers can view all tickets in subscription. Unit Managers can view tickets for IP addresses in their same business unit. Scanners and Readers can view tickets for IP addresses in their own account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ticket_numbers=</td>
<td>(Optional) Specifies ticket numbers for which ticket information will be retrieved. Ticket numbers are integers, assigned by the service automatically. A maximum of 1,000 ticket numbers may be specified. Multiple ticket numbers are comma separated. This parameter or since must be specified.</td>
</tr>
<tr>
<td>since={value}</td>
<td>(Optional) Specifies the start date/time of the time window for retrieving tickets. Only tickets that have been updated within this time window will be retrieved. The end date/time of the time window for retrieving tickets is the date/time when get_tickets.php is run. The start date/time is specified in YYYY-MM-DDTHH:MM:SSZ format (UTC/GMT), like “2005-01-10T02:33:11Z”. This parameter or ticket_numbers must be specified.</td>
</tr>
<tr>
<td>state={value}</td>
<td>(Optional) Specifies the current state of tickets to be retrieved. A valid value is OPEN, RESOLVED, or CLOSED. If unspecified, tickets with all states are retrieved.</td>
</tr>
<tr>
<td>vuln_details={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

Samples

Retrieve remediation tickets that have been updated since July 1, 2018 at 1:00:00 AM (UTC/GMT) and that have any state (Open, Resolved, or Closed):

https://qualysapi.qualys.com/msp/get_tickets.php?
since=2018-07-01T01:00:00Z

Retrieve remediation tickets 002737, 002738, and 002740 with vulnerability details:
The ignore_vuln/index.php function is used to ignore or restore (un-ignore) vulnerabilities on certain hosts. The ignore status applies to a vulnerability/host pair. Vulnerabilities can be set to ignore on hosts so that they do not appear in automatic scan reports, host information reports, asset search reports as well as other views in the Qualys user interface.

Both Vulnerabilities and Potential Vulnerabilities may be set to the ignore status on hosts in the user’s account. Information Gathered issues cannot be set to the ignore status. Note that the following QIDs cannot be set to ignore: 38175 (Unauthorized Service Detected), 82043 (Unauthorized Open Port Detected), 38228 (Required Service Not Detected) and 82051 (Required Port Not Detected).

When making an ignore_vuln/index.php request, you must specify QIDs (up to 10) and target hosts. Host selection parameters allow you to specify hosts by IP address, asset group, asset tag, DNS host name or NetBIOS host name.

**Target Hosts**
A vulnerability can be set to ignore/restore only on hosts with scan results. If a host was previously scanned and then purged, the scan results are removed and no longer available. In this case an ignore vulnerability request will have no effect until a re-scan populates the host with fresh scan results.

The ignore/restore request applies to the target hosts at the time of the request. For example, if you specify an ignore action on asset groups, the request applies to the IP addresses in the asset groups at the time of the request. Subsequently, if an asset group is updated with new IP addresses, the new IPs are not set to the ignore status.

**Ignored Status and Tickets**
The ignore/restore actions have an effect on remediation tickets in the user account. When you set the ignore status for vulnerabilities on hosts, the service closes associated remediation tickets with the ticket state/status of Closed/Ignored. If no ticket exists, a new one will be created and closed automatically for tracking purposes as Closed/Ignored. When you restore vulnerabilities on hosts, the service automatically reopens the associated tickets and sets them to Open/Reopened.
The **ticket_list.php** function allows you to list tickets in the user account and this information could be useful for taking actions using **ignore_vuln/index.php**. For example, you could use **ticket_list.php** to find tickets on certain QIDs in the Closed/Ignored state and then use the information returned to make **ignore_vuln/index.php** requests to restore vulnerabilities on certain hosts.

**Permissions**

User permissions for the **ignore_vuln/index.php** function are described below.

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>Ignore/Restore vulnerabilities and potential vulnerabilities on all hosts in subscription.</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>Ignore/Restore vulnerabilities and potential vulnerabilities on hosts in user’s business unit.</td>
</tr>
<tr>
<td>Scanner</td>
<td>Ignore/Restore vulnerabilities and potential vulnerabilities on hosts in user’s account, when a certain remediation policy option is enabled.*</td>
</tr>
<tr>
<td>Reader</td>
<td>Ignore/Restore vulnerabilities and potential vulnerabilities on hosts in user’s account, when a certain remediation policy option is enabled.*</td>
</tr>
</tbody>
</table>

* Scanners and Readers have permission to ignore/restore vulnerabilities when the option “Allow Scanners and Readers to mark tickets as Closed/Ignored” is enabled in the QualysGuard user interface. A Manager can edit this setting for the subscription. See the QualysGuard online help for information.

**Input Parameters**

The parameters for **ignore_vuln/index.php** are described below.

The request parameters are below:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=ignore</td>
<td>restore</td>
</tr>
<tr>
<td>qids={qid,qid,...}</td>
<td>(Required) Specifies the QIDs (Qualys IDs) to ignore/restore. A maximum of 10 QIDs may be specified. Multiple QIDs are comma separated.</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Required) Specify comments for the action. The comments may include a maximum of 255 characters. Comments are stored with ignored vulnerabilities, and are visible to users in the Qualys user interface.</td>
</tr>
</tbody>
</table>
### VM Remediation Tickets

Set Vulnerabilities to Ignore on Hosts

The host parameters mentioned below are optional and mutually exclusive (only one may be specified per request). At least one parameter must be specified.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reopen_ignored_days={value}</td>
<td>(Optional) Set to reopen ignored vulnerabilities that are detected after a number of days (1-730). If the ignored vulnerability is reopened by the service, the corresponding ticket’s state/status is changed from Closed/Ignored to Open/Reopened.</td>
</tr>
<tr>
<td>reopen_ignored_date={date}</td>
<td>(Optional) Set to reopen ignored vulnerabilities that are detected after a specified date. If the ignored vulnerability is reopened by the service, the corresponding ticket’s state/status is changed from Closed/Ignored to Open/Reopened.</td>
</tr>
<tr>
<td>asset_groups={ag1,ag2,...}</td>
<td>(Optional) Selects hosts by asset group. The hosts included in the one or more asset groups provided are selected. A maximum of 5 asset group titles may be specified. The asset group title &quot;All&quot; as defined in the Qualys user interface may be specified. Multiple asset groups are comma separated. This parameter or another host selection parameter is required.</td>
</tr>
<tr>
<td>ips={nnn, nnn-nnn,...}</td>
<td>(Optional) Selects hosts by IP address. Enter one or more IP addresses and/or ranges. Multiple entries are comma separated. The parameter value may include a maximum of 512 characters (ascii). This parameter or another host selection parameter is required.</td>
</tr>
<tr>
<td>network_id={value}</td>
<td>(Optional) Only valid when the networks feature is enabled. The network ID for the record. This parameter or another host selection parameter is required.</td>
</tr>
<tr>
<td>tag_set_include={value}</td>
<td>(Optional) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. This parameter or another host selection parameter is required.</td>
</tr>
<tr>
<td>tag_set_exclude={value}</td>
<td>(Optional) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. This parameter or another host selection parameter is required.</td>
</tr>
</tbody>
</table>
### VM Remediation Tickets

#### Set Vulnerabilities to Ignore on Hosts

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag_set_by={id</td>
<td>name}</td>
</tr>
<tr>
<td>tag_include_selector= {all</td>
<td>any}</td>
</tr>
<tr>
<td>tag_exclude_selector= {all</td>
<td>any}</td>
</tr>
<tr>
<td>use_ip_nt_range_tags_inclu de={0</td>
<td>1}</td>
</tr>
<tr>
<td>use_ip_nt_range_tags_exclu de={0</td>
<td>1}</td>
</tr>
<tr>
<td>dns_contains={value}</td>
<td>(Optional) Selects hosts by DNS host name. Specify a text string contained in one or more DNS host names. The text string may include a maximum of 100 characters (ascii). This parameter or another host selection parameter is required.</td>
</tr>
<tr>
<td>netbios_contains={value}</td>
<td>(Optional) Selects hosts by NetBIOS host name. Specify a text string contained in one or more NetBIOS host names. The text string may include a maximum of 100 characters (ascii). This parameter or another host selection parameter is required.</td>
</tr>
</tbody>
</table>

### Samples

To ignore QID 19070 “MS-SQL 8.0 UDP Slammer Worm Buffer Overflow Vulnerability” for the hosts in asset group “New York”, use a URL like this:

To restore (un-ignore) QIDs 90305 and 100035 on IP address 10.10.10.33 and IP range 10.10.10.100-10.10.10.120, use a URL like this:

https://qualysapi.qualys.com/api/2.0/fo/ignore_vuln/index.php?action=restore&qids=90305,100035&ips=10.10.10.33,10.10.10.100-10.10.10.120&comments=request+by+GStevenson

If there are ignored vulnerabilities in your account, you can list all ignored vulnerabilities in the account using the ticket_list.php function as shown in the following URL:

https://qualysapi.qualys.com/msp/ticket_list.php?asset_groups=All&states=IGNORED

DTD

<platform API server>/api/2.0/dtd/fo/ignore_vuln_output.dtd
Compliance

Manage compliance policies, exceptions and reports. Policy Compliance (PC) is required.

Compliance Control List
Compliance Policy List
Compliance Policy - Export
Compliance Policy - Import
Compliance Policy - Merge
Compliance Policy - Manage Asset Groups
Compliance Posture Information
PC Posture Streaming APIs
Exceptions
SCAP Cyberscope Report
SCAP ARF Report
SCAP Policy List
Compliance Control List

/api/2.0/fo/compliance/control/?action=list

[GET]  [POST]

View a list of compliance controls which are visible to the user. Controls in the XML output are sorted by control ID in ascending order. Optional input parameters support filtering the list.

Using the Qualys user interface, it’s possible to customize the list of frameworks at the subscription level. Under PC, go to Policies > Setup > Frameworks to customize the frameworks list. If the frameworks list is customized for your subscription, then the customized list of frameworks will appear in the controls list output returned by a control list API request.

Permissions

Note: The Compliance Control APIs are available as part of one of the following subscription combinations only:

- PC and API add-on
- PC, SCA, and API add-on
- VMDR, SCA, and API add-on

Users with PC enabled have the ability to view compliance controls.

Maximum Controls per API Request

The output of the Compliance Control API is paginated. By default, a maximum of 1,000 control records are returned per request. You can customize the page size (i.e. the number of control records) by using the parameter “truncation_limit=2000” for instance. In this case the results will be returned with pages of 2,000 records.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0[1]</td>
<td>(Optional) Show (echo) the request input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.</td>
</tr>
</tbody>
</table>
### Compliance Control List

#### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>details=</td>
<td>Basic</td>
</tr>
<tr>
<td></td>
<td>(Optional) Show the requested amount of information for each control. A valid value is:</td>
</tr>
<tr>
<td></td>
<td>None - show control ID only</td>
</tr>
<tr>
<td></td>
<td>Basic (default) - show control ID and basic control information: the control category, sub-category, statement, and technology information</td>
</tr>
<tr>
<td></td>
<td>All - show control ID, basic control information, and framework mappings</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Optional) Show only certain control IDs and/or ID ranges. Multiple entries are comma separated. One or more control IDs/ranges may be specified. A control ID range entry is specified with a hyphen (for example, 3000-3250). Valid control IDs are required.</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(Optional) Show only controls which have a minimum control ID value. A valid control ID is required.</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(Optional) Show only controls which have a maximum control ID value. A valid control ID is required.</td>
</tr>
<tr>
<td>updated_after_datetime={value}</td>
<td>(Optional) Show only controls updated after a certain date/time. See “Date Filters” below.</td>
</tr>
<tr>
<td>created_after_datetime={value}</td>
<td>(Optional) Show only controls created after a certain date/time. See “Date Filters” below.</td>
</tr>
<tr>
<td>truncation_limit={value}</td>
<td>(Optional) The maximum number of control records processed per request. When not specified, the truncation limit is set to 1,000 host records. You may specify a value less than the default (1-999) or greater than the default (1001-1000000).</td>
</tr>
<tr>
<td></td>
<td>If the requested list identifies more records than the truncation limit, then the XML output includes the &lt;WARNING&gt; element and the URL for making another request for the next batch of records.</td>
</tr>
<tr>
<td></td>
<td>You can specify truncation_limit=0 for no truncation limit. This means that the output is not paginated and all the records are returned in a single output. WARNING: This can generate very large output and processing large XML files can consume a lot of resources on the client side. In this case it is recommended to use the pagination logic and parallel processing. The previous page can be processed while the next page is being downloaded.</td>
</tr>
</tbody>
</table>

#### Date Filters

The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2010-03-01” or “2010-03-01T23:12:00Z”

If you specify a date but no time as for example 2010-03-01, then the service automatically sets the time to 2010-03-01T00:00:00Z (the start of the day).
When date filters are specified using both input parameters for a single API request, both date filters are satisfied (ANDed).

**DTD**

```
<platform API server>/api/2.0/fo/compliance/control/control_list_output.dtd
```

**Sample - Control List Output**

This sample control list output was produced for CID 1044 with details=Basic.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM "https://qualyspapi.qualys.com/api/2.0/fo/compliance/control/control_list_output.dtd">

<CONTROL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2010-03-16T22:53:05Z</DATETIME>
    <CONTROL_LIST>
      <CONTROL>
        <ID>1044</ID>
        <UPDATE_DATE>2010-02-12T00:00:00Z</UPDATE_DATE>
        <CREATED_DATE>2007-10-12T00:00:00Z</CREATED_DATE>
        <CATEGORY>Access Control Requirements</CATEGORY>
        <SUBCATEGORY><![CDATA[Authorizations (Multi-user ACL/role)]]></SUBCATEGORY>
        <STATEMENT><![CDATA['O7_DICTIONARY_ACCESSIBILITY' setting in init.ora (ORACLE Data Dictionary)]]></STATEMENT>
        <TECHNOLOGY_LIST>
          <TECHNOLOGY>
            <ID>7</ID>
            <NAME>Oracle 9i</NAME>
            <RATIONALE><![CDATA[The "O7_DICTIONARY_ACCESSIBILITY" setting allows control/restrictions to be placed on the user's SYSTEM privileges. If this parameter is set to TRUE, SYS schema access will be allowed, which is the default for Oracle operations. Restricting this system privilege with a setting of FALSE will allow users or roles granted SELECT ANY TABLE access to objects in the normal schema, but disallow access to objects in the SYS schema, unless access is specifically granted.]]></RATIONALE>
          </TECHNOLOGY>
          <TECHNOLOGY>
            <ID>8</ID>
            <NAME>Oracle 10g</NAME>
            <RATIONALE><![CDATA[The "O7_DICTIONARY_ACCESSIBILITY" setting allows control/restrictions to be placed on the user's SYSTEM privileges. If this parameter is set to TRUE, SYS schema access...]]></RATIONALE>
          </TECHNOLOGY>
        </TECHNOLOGY_LIST>
      </CONTROL>
    </CONTROL_LIST>
  </RESPONSE>
</CONTROL_LIST_OUTPUT>
```
access will be allowed, which is the default for Oracle operations. Restricting this system privilege with a setting of FALSE will allow users or roles granted SELECT ANY TABLE access to objects in the normal schema, but disallow access to objects in the SYS schema, unless access is specifically granted.]]>
</RATIONALITY>
</TECHNOLOGY>
</TECHNOLOGY_LIST>
</CONTROL>

<CONTROL>
<ID>1045</ID>
<UPDATE_DATE>2010-03-03T00:00:00Z</UPDATE_DATE>
<CREATED_DATE>2007-10-12T00:00:00Z</CREATED_DATE>
<CATEGORY>OS Security Settings</CATEGORY>
<SUBCATEGORY><![CDATA[System Settings (OSI layers 6-7)]]>

<STATEMENT><![CDATA[Status of the 'Clipbook' service (Guidance = Disabled)]]>
</STATEMENT>

<TECHNOLOGY_LIST>
<TECHNOLOGY>
<ID>1</ID>
<NAME>Windows XP desktop</NAME>
<RATIONALITY><![CDATA[The 'Clipbook' service is used to transfer Clipboard information across the LAN and is sent in clear text. The authentication required is a holdover from the 16-bit 'Network Dynamic Data Exchange' protocol, which is a 'network' password among systems sharing the LAN, with a default set allow READ for EVERYONE that has network access. As this Windows service is not required for any other system operations and increases system vulnerability it should be disabled unless there is a demonstrated need for its use set by the business.]]>
</RATIONALITY>
</TECHNOLOGY>
<TECHNOLOGY>
<ID>2</ID>
<NAME>Windows 2003 Server</NAME>
<RATIONALITY><![CDATA[The 'Clipbook' service is used to transfer Clipboard information across the LAN and is sent in clear text. The authentication required is a holdover from the 16-bit 'Network Dynamic Data Exchange' protocol, which is a 'network' password among systems sharing the LAN, with a default set allow READ for EVERYONE that has network access. As this Windows service is not required for any other system operations and increases system vulnerability it should be disabled unless there is a demonstrated need for its use set by the business.]]>
</RATIONALITY>
</TECHNOLOGY>
</TECHNOLOGY_LIST>
</CONTROL>
transfer Clipboard information across the LAN and is sent in clear text. The authentication required is a holdover from the 16-bit 'Network Dynamic Data Exchange' protocol, which is a 'network' password among systems sharing the LAN, with a default set allow READ for EVERYONE that has network access. As this Windows service is not required for any other system operations and increases system vulnerability it should be disabled unless there is a demonstrated need for its use set by the business.]]></RATIONALE>

</RESPONSE>
</CONTROL_LIST_OUTPUT>

Updates you'll see once Agent UDC support is available

New Agent UDC Support will be announced soon via the Qualys Technology blog once remaining components are released.

The XML output may include the USE_AGENT_ONLY element for these Windows and Unix control types: Directory Search Control and Directory Integrity Control. This is set to 1 when the “Use agent scan only” option is enabled for the control.

The XML output may include the AUTO_UPDATE element for these Windows and Unix control types: File Integrity Control and Directory Integrity Control. This is set to 1 when the “Auto update expected value” option is enabled for the control.

Option to disable the case-sensitive search in Unix agent UDCs (Directory Search and Directory Integrity) is available. Once the <DISABLE_CASE_SENSITIVE_SEARCH> parameter is enabled (true), the search result lists all possible combinations in the upper and/or lower case file name. By default, this option is disabled (false) which lists result with case-sensitive file name.

Sample - Control List Output when Agent UDC Support is available

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/control/control_list_output.dtd">
<CONTROL_LIST_OUTPUT>
<RESPONSE>
Database UDC for MS SQL, Oracle, Sybase, PostgreSQL/Pivotal Greenplum, SAP IQ, and IBM DB2

You can create custom controls for MSSQL, Oracle, Sybase, PostgreSQL/Pivotal Greenplum, SAP IQ, and IBM DB2 databases. To support database controls, we've added new elements to the XML output and DTDs for Control List Output and Policy Export Output.

Sample - Control List API for MS SQL

**API request:**
```
curl -u "username:password" -H "Content-type: text/xml" -X "POST" 
-d "action=list&details=All&ids=100022" 
"https://qualysapi.qualys.com/api/2.0/fo/compliance/control/" 
MSSQLControlAPI.xml
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM 
"https://qualysapi.qualys.com/api/2.0/fo/compliance/control/control_list_output.dtd"> 
<CONTROL_LIST_OUTPUT> 
<RESPONSE> 
<DATETIME>2019-05-08T18:31:17Z</DATETIME> 
<CONTROL_LIST> 
<CONTROL> 
<ID>100022</ID> 
<UPDATE_DATE>2019-05-08T18:31:08Z</UPDATE_DATE> 
```
<CREATED_DATE>2019-04-29T20:21:11Z</CREATED_DATE>
<CATEGORY>Access Control Requirements</CATEGORY>
<STATEMENT>&lt;[CDATA[CustomerData]]&gt;</STATEMENT>
<CRI

<DATETIME>2019-05-08T18:32:46Z</DATETIME>
<CONTROL_LIST>
  <CONTROL>
    <ID>100060</ID>
    <UPDATE_DATE>2019-05-08T18:32:04Z</UPDATE_DATE>
    <CREATED_DATE>2019-05-03T19:32:18Z</CREATED_DATE>
    <CATEGORY>Database Settings</CATEGORY>
    <SUB_CATEGORY><![CDATA[DB Access Controls]]></SUB_CATEGORY>
    <STATEMENT><![CDATA[OracleselectAllCustomerData]]></STATEMENT>
    <CRITICALITY>
      <LABEL><![CDATA[MINIMAL]]></LABEL>
      <VALUE>1</VALUE>
    </CRITICALITY>
    <CHECK_TYPE><![CDATA[Oracle Database Check]]></CHECK_TYPE>
    <COMMENT><![CDATA[Gather All Data]]></COMMENT>
    <IGNORE_ERROR>1</IGNORE_ERROR>
    <ERROR_SET_STATUS>FAIL</ERROR_SET_STATUS>
    <TECHNOLOGY_LIST>
      <TECHNOLOGY>
        <ID>7</ID>
        <NAME>Oracle 9i</NAME>
        <RATIONALE><![CDATA[GatherAllData]]></RATIONALE>
        <DB_QUERY><![CDATA[SELECT * FROM Customers WHERE ROWNUM >= 3;]]></DB_QUERY>
        <DESCRIPTION><![CDATA[select all the data]]></DESCRIPTION>
      </TECHNOLOGY>
      <TECHNOLOGY>
        <ID>8</ID>
        <NAME>Oracle 10g</NAME>
        <RATIONALE><![CDATA[GatherAllData]]></RATIONALE>
        <DB_QUERY><![CDATA[SELECT * FROM Customers;]]></DB_QUERY>
        <DESCRIPTION><![CDATA[select all the data]]></DESCRIPTION>
      </TECHNOLOGY>
      ...</TECHNOLOGY>
    </TECHNOLOGY_LIST>
  </CONTROL>
</CONTROL_LIST_OUTPUT>
Sample - Control List API for Sybase

API request:
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
-d "action=list&details=All&ids=100947"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/control/

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/control/control_list_output.dtd">
<CONTROL_LIST_OUTPUT>
  <REQUEST>
    <DATETIME>2020-03-21T05:29:10Z</DATETIME>
    <USER_LOGIN>quays_sp1</USER_LOGIN>
    <RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/compliance/control/</RESOURCE>
    <PARAM_LIST>
      <PARAM>
        <KEY>action</KEY>
        <VALUE>list</VALUE>
      </PARAM>
      <PARAM>
        <KEY>ids</KEY>
        <VALUE>100947</VALUE>
      </PARAM>
      <PARAM>
        <KEY>echo_request</KEY>
        <VALUE>1</VALUE>
      </PARAM>
    </PARAM_LIST>
  </REQUEST>
  <RESPONSE>
    <DATETIME>2020-03-21T05:29:10Z</DATETIME>
    <CONTROL_LIST>
      <CONTROL>
        <ID>100947</ID>
        <UPDATE_DATE>2020-03-20T15:05:35Z</UPDATE_DATE>
        <CREATED_DATE>2020-03-18T05:50:27Z</CREATED_DATE>
        <CATEGORY>Access Control Requirements</CATEGORY>
        <SUB_CATEGORY><![CDATA[Account Creation/User Management]]></SUB_CATEGORY>
        <STATEMENT>
<![CDATA[sybase db udc]]>
</STATEMENT>
<CRITICALITY>
<label>
<![CDATA[UNDEFINED]]>
</label>
<VALUE>0</VALUE>
</CRITICALITY>
<CHECK_TYPE>
<![CDATA[Sybase Database Check]]>
</CHECK_TYPE>
<COMMENT>
<![CDATA[null]]>
</COMMENT>
<IGNORE_ERROR>0</IGNORE_ERROR>
<ERROR_SET_STATUS/>
<TECHNOLOGY_LIST>
<TECHNOLOGY>
<ID>69</ID>
<NAME>Sybase ASE 15</NAME>
<RATIONALE>
<![CDATA[select db_name() as dbname, s.name as segment_name, t.free_space as free_space_pages,
case t.status when 1 then 'LAST CHANCE' else 'OTHER' end as status,
t.proc_name, suser_name(t.suid) as owner
from syssegments s, systhresholds t
where t.segment = s.segment]]>
</RATIONALER>
</DB_QUERY>
</DESCRIPTION>
<![CDATA[select db_name() as dbname, s.name as segment_name, t.free_space as free_space_pages,
case t.status when 1 then 'LAST CHANCE' else 'OTHER' end as status,
t.proc_name, suser_name(t.suid) as owner
from syssegments s, systhresholds t
where t.segment = s.segment]]>
</DB_QUERY>
<DESCRIPTION>
<![CDATA[select db_name() as dbname, s.name as segment_name, t.free_space as free_space_pages,
case t.status when 1 then 'LAST CHANCE' else 'OTHER' end as status,
t.proc_name, suser_name(t.suid) as owner
from syssegments s, systhresholds t
where t.segment = s.segment]]>
</DESCRIPTION>
from syssegments s, systhresholds t
where t.segment = s.segment]
</DESCRIPTION>
</TECHNOLOGY>
<TECHNOLOGY>
<ID>116</ID>
<NAME>SAP Adaptive Server Enterprise 16</NAME>
<RATIONALE>
<![CDATA[select db_name() as dbname,
  s.name as segment_name,
  t.free_space as free_space_pages,
  case t.status when 1 then 'LAST CHANCE' else 'OTHER' end as status,
  t.proc_name, suser_name(t.suid) as owner
from syssegments s, systhresholds t
where t.segment = s.segment]]>
</RATIONALE>
<DB_QUERY>
<![CDATA[select db_name() as dbname,
  s.name as segment_name,
  t.free_space as free_space_pages,
  case t.status when 1 then 'LAST CHANCE' else 'OTHER' end as status,
  t.proc_name, suser_name(t.suid) as owner
from syssegments s, systhresholds t
where t.segment = s.segment]]>
</DB_QUERY>
<DESCRIPTION>
<![CDATA[select db_name() as dbname,
  s.name as segment_name,
  t.free_space as free_space_pages,
  case t.status when 1 then 'LAST CHANCE' else 'OTHER' end as status,
  t.proc_name, suser_name(t.suid) as owner
from syssegments s, systhresholds t
where t.segment = s.segment]]>
</DESCRIPTION>
</TECHNOLOGY>
</TECHNOLOGY_LIST>
</CONTROL>
</CONTROL_LIST>
</RESPONSE>
</CONTROL_LIST_OUTPUT>
Sample - Control List API for PostgreSQL/Pivotal Greenplum

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=list&details=All&ids=101335"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/control/
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/control/control_list_output.dtd">
<CONTROL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-10-15T16:59:13Z</DATETIME>
    <CONTROL_LIST>
      <CONTROL>
        <ID>101335</ID>
        <UPDATE_DATE>2020-10-14T20:11:29Z</UPDATE_DATE>
        <CREATED_DATE>2020-10-14T19:46:01Z</CREATED_DATE>
        <CATEGORY>Access Control Requirements</CATEGORY>
        <SUB_CATEGORY><![CDATA[Account Creation/User Management]]></SUB_CATEGORY>
        <STATEMENT><![CDATA[prePostGreSQL_selectStatement]]></STATEMENT>
        <CRITICALITY>
          <LABEL><![CDATA[URGENT]]></LABEL>
          <VALUE>5</VALUE>
        </CRITICALITY>
        <CHECK_TYPE><![CDATA[PostgreSQL Database Check]]></CHECK_TYPE>
        <COMMENT><![CDATA[comments]]></COMMENT>
        <IGNORE_ERROR>0</IGNORE_ERROR>
        <ERROR_SET_STATUS></ERROR_SET_STATUS>
      </CONTROL>
      <TECHNOLOGY>
        <ID>114</ID>
        <NAME>PostgreSQL 9.x</NAME>
        <RATIONALE><![CDATA[Rationale]]></RATIONALE>
        <DB_QUERY><![CDATA[select name, setting from pg_catalog.pg_settings where name='log_min_duration_statement']]]></DB_QUERY>
        <DESCRIPTION><![CDATA[Description]]></DESCRIPTION>
      </TECHNOLOGY>
    </CONTROL_LIST>
  </RESPONSE>
</CONTROL_LIST_OUTPUT>
```
<ID>143</ID>
<NAME>PostgreSQL 10.x</NAME>
<RATIONALE><![CDATA[Rationale]]></RATIONALE>
<DB_QUERY><![CDATA[select name, setting from pg_catalog.pg_settings where name='log_min_duration_statement']]></DB_QUERY>
<DESCRIPTION><![CDATA[Description]]></DESCRIPTION>
</TECHNOLOGY>

<TECHNOLOGY>
<ID>192</ID>
<NAME>PostgreSQL 11.x</NAME>
<RATIONALE><![CDATA[Rationale]]></RATIONALE>
<DB_QUERY><![CDATA[select name, setting from pg_catalog.pg_settings where name='log_min_duration_statement']]></DB_QUERY>
<DESCRIPTION><![CDATA[Description]]></DESCRIPTION>
</TECHNOLOGY>

<TECHNOLOGY>
<ID>201</ID>
<NAME>Pivotal Greenplum 5.x</NAME>
<RATIONALE><![CDATA[Rationale]]></RATIONALE>
<DB_QUERY><![CDATA[select name, setting from pg_catalog.pg_settings where name='log_min_duration_statement']]></DB_QUERY>
<DESCRIPTION><![CDATA[Description]]></DESCRIPTION>
</TECHNOLOGY>

<TECHNOLOGY>
<ID>228</ID>
<NAME>PostgreSQL 12.x</NAME>
<RATIONALE><![CDATA[Rationale]]></RATIONALE>
<DB_QUERY><![CDATA[select name, setting from pg_catalog.pg_settings where name='log_min_duration_statement']]></DB_QUERY>
<DESCRIPTION><![CDATA[Description]]></DESCRIPTION>
</TECHNOLOGY>

<TECHNOLOGY>
>ID>230</ID>
<NAME>Pivotal Greenplum 6.x</NAME>
<RATIONALE><![CDATA[Rationale]]></RATIONALE>
<DB_QUERY><![CDATA[select name, setting from pg_catalog.pg_settings where name='log_min_duration_statement']]></DB_QUERY>
<DESCRIPTION><![CDATA[Description]]></DESCRIPTION>
</TECHNOLOGY>
</TECHNOLOGY_LIST>
Sample - Control List API for IBM DB2

API Request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "action=list&ids=100010" "https://qualysapi.qualys.com/api/2.0/fo/compliance/control/
```

XML Output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/control/control_list_output.dtd">
<CONTROL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-06-22T11:14:08Z</DATETIME>
    <CONTROL_LIST>
      <CONTROL>
        <ID>100010</ID>
        <UPDATE_DATE>2021-06-22T08:24:27Z</UPDATE_DATE>
        <CREATED_DATE>2021-06-22T08:24:27Z</CREATED_DATE>
        <CATEGORY>Database Settings</CATEGORY>
        <SUBCATEGORY><![CDATA[DB Access Controls]]></SUBCATEGORY>
        <STATEMENT><![CDATA[db2 statement]]></STATEMENT>
        <CRITICALITY>
          <LABEL><![CDATA[SERIOUS]]></LABEL>
          <VALUE>3</VALUE>
        </CRITICALITY>
        <CHECK_TYPE><![CDATA[DB2 Database Check]]></CHECK_TYPE>
        <COMMENT><![CDATA[comment for db2 udc]]></COMMENT>
        <IGNORE_ERROR>1</IGNORE_ERROR>
        <ERROR_SET_STATUS>FAIL</ERROR_SET_STATUS>
        <TECHNOLOGY_LIST>
          <TECHNOLOGY>
            <ID>40</ID>
            <NAME>IBM DB2 9.x</NAME>
            <RATIONALE><![CDATA[db2 udc rationale]]></RATIONALE>
            <DB_QUERY><![CDATA[select * from sysadmin]]></DB_QUERY>
            <DESCRIPTION><![CDATA[test db2 udc description]]></DESCRIPTION>
          </TECHNOLOGY>
        </TECHNOLOGY_LIST>
      </CONTROL>
    </CONTROL_LIST>
  </RESPONSE>
</CONTROL_LIST_OUTPUT>
```
Sample - Control List API for File Content Check

**API request:**
```
curl -u "username:password" -H "Content-type: text/xml" -X "POST" -d "action=list&echo_request=1&ids=100006,100000,100026&details=All" "https://qualysapi.qualys.com/api/2.0/fo/compliance/control/">control_list.xml
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/control/control_list_output.dtd">
<CONTROL_LIST_OUTPUT>

<REQUEST>
<DATETIME>2019-10-14T21:17:21Z</DATETIME>
<USER_LOGIN>username</USER_LOGIN>
<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/compliance/control/</RESOURCE>

<PARAM_LIST>
<PARAM>
```
```
<KEY>action</KEY>
   <VALUE>list</VALUE>
</PARAM>
<PARAM>
   <KEY>echo_request</KEY>
   <VALUE>1</VALUE>
</PARAM>
<PARAM>
   <KEY>ids</KEY>
   <VALUE>100006,100000,100026</VALUE>
</PARAM>
<PARAM>
   <KEY>details</KEY>
   <VALUE>All</VALUE>
</PARAM>
</PARAM_LIST>
</REQUEST>

<RESPONSE>
<DATETIME>2019-10-14T21:17:21Z</DATETIME>

<CONTROL_LIST>
<CONTROL>
   <ID>100000</ID>
   <UPDATE_DATE>2019-10-10T21:54:35Z</UPDATE_DATE>
   <CREATED_DATE>2019-10-08T19:16:02Z</CREATED_DATE>
   <CATEGORY>Access Control Requirements</CATEGORY>
   <SUBCATEGORY><![CDATA[Account Creation/User Management]]></SUBCATEGORY>
   <STATEMENT><![CDATA[preFCCUDC]]></STATEMENT>
   <CRITICALITY>
      <LABEL><![CDATA[min]]></LABEL>
      <VALUE>1</VALUE>
   </CRITICALITY>
   <CHECK_TYPE><![CDATA[Windows File Content Check]]></CHECK_TYPE>
   <COMMENT><![CDATA[]]></COMMENT>
   <IGNORE_ERROR>0</IGNORE_ERROR>
   <IGNORE_ITEM_NOT_FOUND>0</IGNORE_ITEM_NOT_FOUND>
   <SCAN_PARAMETERS>
      <PATH_TYPE><![CDATA[Use file search]]></PATH_TYPE>
      <FILE_QUERY><![CDATA[QWEB*]]></FILE_QUERY>
      <BASE_DIR><![CDATA[c:\]]></BASE_DIR>
      <DEPTH_LIMIT><![CDATA[3]]></DEPTH_LIMIT>
      <FILE_NAME_MATCH><![CDATA[preTest2.txt]]></FILE_NAME_MATCH>
      <FILE_NAME_SKIP><![CDATA[]]></FILE_NAME_SKIP>
      <DIR_NAME_MATCH><![CDATA[*]]></DIR_NAME_MATCH>
   </SCAN_PARAMETERS>
</CONTROL>
</CONTROL_LIST>
</RESPONSE>
<COMMENT><![CDATA[]]></COMMENT>
<IGNORE_ERROR>0</IGNORE_ERROR>
<IGNORE_ITEM_NOT_FOUND>0</IGNORE_ITEM_NOT_FOUND>
<SCAN_PARAMETERS>
  <PATH_TYPE><![CDATA[Use Registry key]]></PATH_TYPE>
  <REG_HIVE><![CDATA[HKEY_CLASSES_ROOT (HKCR)]]]></REG_HIVE>
  <REG_KEY><![CDATA[TestKey\user]]></REG_KEY>
  <REG_VALUE_NAME><![CDATA[preName]]></REG_VALUE_NAME>
  <FILE_PATH><![CDATA[]]></FILE_PATH>
  <FILE_QUERY><![CDATA[.*]]></FILE_QUERY>
  <DATA_TYPE>String List</DATA_TYPE>
  <DESCRIPTION><![CDATA[reg key]]></DESCRIPTION>
</SCAN_PARAMETERS>
<TECHNOLOGY_LIST>
  <TECHNOLOGY>
    <ID>53</ID>
    <NAME>Windows 2012 Server</NAME>
    <RATIONALE><![CDATA[rationale]]></RATIONALE>
    <DATAPOINT>
      <CARDINALITY>contains</CARDINALITY>
      <OPERATOR>xre</OPERATOR>
      <DEFAULT_VALUES total="1">
        <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
      </DEFAULT_VALUES>
    </DATAPOINT>
  </TECHNOLOGY>
  <TECHNOLOGY>
    <ID>75</ID>
    <NAME>Windows Server 2012 R2</NAME>
    <RATIONALE><![CDATA[rationale]]></RATIONALE>
    <DATAPOINT>
      <CARDINALITY>contains</CARDINALITY>
      <OPERATOR>xre</OPERATOR>
      <DEFAULT_VALUES total="1">
        <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
      </DEFAULT_VALUES>
    </DATAPOINT>
  </TECHNOLOGY>
</TECHNOLOGY_LIST>
</CONTROL>
<CONTROL>
  <ID>100026</ID>
  <UPDATE_DATE>2019-10-11T20:12:48Z</UPDATE_DATE>
  <CREATED_DATE>2019-10-11T20:12:48Z</CREATED_DATE>
  <CATEGORY>Access Control Requirements</CATEGORY>
</CONTROL>
<SUBCATEGORY><![CDATA[Account Creation/User Management]]></SUBCATEGORY>

<STATEMENT><![CDATA[pre_fcc_file_path_regexwith$]]></STATEMENT>
  <CRITICALITY>
    <LABEL><![CDATA[min]]></LABEL>
    <VALUE>1</VALUE>
  </CRITICALITY>
  <CHECK_TYPE><![CDATA[Windows File Content Check]]></CHECK_TYPE>
  <COMMENT><![CDATA[]]></COMMENT>
  <IGNORE_ERROR>0</IGNORE_ERROR>
  <IGNORE_ITEM_NOT_FOUND>0</IGNORE_ITEM_NOT_FOUND>
  <SCAN_PARAMETERS>
    <PATH_TYPE><![CDATA[Use file path]]></PATH_TYPE>
    <FILE_PATH><![CDATA[C:\user\PreTest\pretestfile1.txt]]></FILE_PATH>
      <FILE_QUERY><![CDATA[pre$]]></FILE_QUERY>
    </SCAN_PARAMETERS>
  </SCAN_PARAMETERS>
  <TECHNOLOGY_LIST>
    <TECHNOLOGY>
      <ID>1</ID>
      <NAME>Windows XP desktop</NAME>
      <RATIONALE><![CDATA[ration]]></RATIONALE>
      <DATAPoint>
        <CARDINALITY>contains</CARDINALITY>
        <OPERATOR>xre</OPERATOR>
        <DEFAULT_VALUES total="1">
          <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
        </DEFAULT_VALUES>
      </DATAPoint>
    </TECHNOLOGY>
    <TECHNOLOGY>
      <ID>2</ID>
      <NAME>Windows 2003 Server</NAME>
      <RATIONALE><![CDATA[ration]]></RATIONALE>
      <DATAPoint>
        <CARDINALITY>contains</CARDINALITY>
        <OPERATOR>xre</OPERATOR>
        <DEFAULT_VALUES total="1">
          <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
        </DEFAULT_VALUES>
      </DATAPoint>
    </TECHNOLOGY>
  </TECHNOLOGY_LIST>
<TECHNOLOGY>
  <ID>12</ID>
  <NAME>Windows 2000</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
    <OPERATOR>xre</OPERATOR>
    <DEFAULT VALUES total="1">
      <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
    </DEFAULT_VALUES>
  </DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>18</ID>
  <NAME>Windows Vista</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
    <OPERATOR>xre</OPERATOR>
    <DEFAULT VALUES total="1">
      <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
    </DEFAULT_VALUES>
  </DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>21</ID>
  <NAME>Windows 2008 Server</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
    <OPERATOR>xre</OPERATOR>
    <DEFAULT_VALUES total="1">
      <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
    </DEFAULT_VALUES>
  </DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>37</ID>
  <NAME>Windows 7</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
    <OPERATOR>xre</OPERATOR>
    <DEFAULT_VALUES total="1">
      <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
    </DEFAULT_VALUES>
  </DATAPoint>
</TECHNOLOGY>
<DEFAULT_VALUES>
</DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>53</ID>
  <NAME>Windows 2012 Server</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
    <OPERATOR>xre</OPERATOR>
    <DEFAULT_VALUES total="1">
      <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
    </DEFAULT_VALUES>
  </DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>54</ID>
  <NAME>Windows 8</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
    <OPERATOR>xre</OPERATOR>
    <DEFAULT_VALUES total="1">
      <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
    </DEFAULT_VALUES>
  </DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>72</ID>
  <NAME>Windows 8.1</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
    <OPERATOR>xre</OPERATOR>
    <DEFAULT_VALUES total="1">
      <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
    </DEFAULT_VALUES>
  </DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>75</ID>
  <NAME>Windows Server 2012 R2</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
  </DATAPoint>
</TECHNOLOGY>
<OPERATOR>xre</OPERATOR>
<DEFAULT_VALUES total="1">
  <DEFAULT_VALUE><![CDATA[*]]></DEFAULT_VALUE>
</DEFAULT_VALUES>
</DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>91</ID>
  <NAME>Windows 10</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
    <OPERATOR>xre</OPERATOR>
    <DEFAULT_VALUES total="1">
      <DEFAULT_VALUE><![CDATA[*]]></DEFAULT_VALUE>
    </DEFAULT_VALUES>
  </DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>106</ID>
  <NAME>Windows 2016 Server</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
    <OPERATOR>xre</OPERATOR>
    <DEFAULT_VALUES total="1">
      <DEFAULT_VALUE><![CDATA[*]]></DEFAULT_VALUE>
    </DEFAULT_VALUES>
  </DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>144</ID>
  <NAME>Windows Embedded 7</NAME>
  <RATIONALE><![CDATA[ration]]></RATIONALE>
  <DATAPoint>
    <CARDINALITY>contains</CARDINALITY>
    <OPERATOR>xre</OPERATOR>
    <DEFAULT_VALUES total="1">
      <DEFAULT_VALUE><![CDATA[*]]></DEFAULT_VALUE>
    </DEFAULT_VALUES>
  </DATAPoint>
</TECHNOLOGY>

<TECHNOLOGY>
  <ID>145</ID>
  <NAME>Windows Embedded 8</NAME>
<RATIONALITY><![CDATA[ration]]></RATIONALITY>
<DataPoint>
  <Cardinality>contains</Cardinality>
  <Operator>xre</Operator>
  <DefaultValues total="1">
    <DefaultValue>![CDATA[.*]]></DefaultValue>
  </DefaultValues>
</DataPoint>
</Technology>
<Technology>
  <ID>146</ID>
  <Name>Windows Embedded 8.1</Name>
  <Rationale><![CDATA[ration]]></Rationale>
  <DataPoint>
    <Cardinality>contains</Cardinality>
    <Operator>xre</Operator>
    <DefaultValues total="1">
      <DefaultValue>![CDATA[.*]]></DefaultValue>
    </DefaultValues>
  </DataPoint>
</Technology>
<Technology>
  <ID>180</ID>
  <Name>Windows 2019 Server</Name>
  <Rationale><![CDATA[ration]]></Rationale>
  <DataPoint>
    <Cardinality>contains</Cardinality>
    <Operator>xre</Operator>
    <DefaultValues total="1">
      <DefaultValue>![CDATA[.*]]></DefaultValue>
    </DefaultValues>
  </DataPoint>
</Technology>
List Unix File Content Custom Controls when Evaluate as string is enabled

You have an option in Unix File Content custom controls to evaluate scan results as a string instead of string list. Once the `<EVALUATE_AS_STRING>` parameter is enabled (1), the scan result is evaluated as a single string. By default the option is disabled (0).

Sample: List FC UDC when Evaluate as string is enabled

API Request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d "action=list&ids=102090&details=All" "https://qualysapi.qualys.com/api/2.0/fo/compliance/control/
```

XML Output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/control/control_list_output.dtd">
<CONTROL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-04-06T11:14:08Z</DATETIME>
    <CONTROL_LIST>
      <CONTROL>
        <ID>102090</ID>
        <UPDATE_DATE>2021-04-01T11:59:40Z</UPDATE_DATE>
        <CREATED_DATE>2021-04-01T11:59:40Z</CREATED_DATE>
        <CATEGORY>Web Application Services</CATEGORY>
        <SUB_CATEGORY><![CDATA[Web Server/Tier Settings]]></SUB_CATEGORY>
        <STATEMENT><![CDATA[FC_New Option Enabled _With String list]]></STATEMENT>
        <CRITICALITY>
          <LABEL><![CDATA[URGENT]]></LABEL>
          <VALUE>5</VALUE>
        </CRITICALITY>
        <CHECK_TYPE><![CDATA[Unix File Content Check]]></CHECK_TYPE>
        <COMMENT><![CDATA[String list]]></COMMENT>
        <IGNORE_ERROR>1</IGNORE_ERROR>
        <IGNORE_ITEM_NOT_FOUND>1</IGNORE_ITEM_NOT_FOUND>
        <SCAN_PARAMETERS>
          <FILE_PATH><![CDATA[/home/testscan/samram]]></FILE_PATH>
          <FILE_QUERY><![CDATA[.*]]></FILE_QUERY>
          <DATA_TYPE>String List</DATA_TYPE>
          <EVALUATE_AS_STRING>1</EVALUATE_AS_STRING>
          <DESCRIPTION><![CDATA[with string list]]></DESCRIPTION>
```
Sample - List DS UDCs when case sensitive search is disabled

API Request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d "action=list&ids=102154&details=All" "https://qualysapi.qualys.com/api/2.0/fo/compliance/control/

XML Output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/control/control_list_output.dtd">
<CONTROL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-07-21T12:14:26Z</DATETIME>
    <CONTROL_LIST>
      <CONTROL>
        <ID>102154</ID>
        <UPDATE_DATE>2021-07-21T07:02:43Z</UPDATE_DATE>
        <CREATED_DATE>2021-07-07T06:38:30Z</CREATED_DATE>
        <CATEGORY>Access Control Requirements</CATEGORY>
        <SUBCATEGORY>![CDATA[Account Creation/User Management]]></SUBCATEGORY>
        <STATEMENT>![CDATA[DS UDC case sensitive with new option]]></STATEMENT>
        <CRITICALITY>
          <LABEL>![CDATA[M INIMAL]]></LABEL>
          <VALUE>1</VALUE>
        </CRITICALITY>
        <CHECK_TYPE>![CDATA[Unix Directory Search Check]]></CHECK_TYPE>
        <COMMENT>![CDATA[DI UDC case sensitive disabled]]></COMMENT>
        <USE_AGENT_ONLY>1</USE_AGENT_ONLY>
        <IGNORE_ERROR>0</IGNORE_ERROR>
        <SCAN_PARAMETERS>
          <BASE_DIR>![CDATA[/home/qa]]></BASE_DIR>
          <SHOULD_DESCEND>![CDATA[true]]></SHOULD_DESCEND>
          <DEPTH_LIMIT>![CDATA[10]]></DEPTH_LIMIT>
          <FOLLOW_SYMLINK>![CDATA[true]]></FOLLOW_SYMLINK>
          <FILE_NAME_MATCH>![CDATA[*]]></FILE_NAME_MATCH>
          <DIR_NAME_MATCH>![CDATA[*]]></DIR_NAME_MATCH>
        </SCAN_PARAMETERS>
      </CONTROL>
    </CONTROL_LIST>
  </RESPONSE>
</CONTROL_LIST_OUTPUT>
<PERMISSIONS>

<USER>
  <READ>any</READ>
  <WRITE>any</WRITE>
  <EXECUTE>any</EXECUTE>
</USER>

<GROUP>
  <READ>any</READ>
  <WRITE>any</WRITE>
  <EXECUTE>any</EXECUTE>
</GROUP>

<OTHER>
  <READ>any</READ>
  <WRITE>any</WRITE>
  <EXECUTE>any</EXECUTE>
</OTHER>
</PERMISSIONS>

<PERM_COND><![CDATA[all]]></PERM_COND>

<TYPE_MATCH><![CDATA[d,f,l,p,b,c,s,D]]></TYPE_MATCH>

<User_OWNER><![CDATA[Any User]]></User_OWNER>

<Group_OWNER><![CDATA[Any Group]]></Group_OWNER>

<TIME_LIMIT><![CDATA[300]]></TIME_LIMIT>

<MATCH_LIMIT><![CDATA[50]]></MATCH_LIMIT>

<DISABLE_CASE_SENSITIVE_SEARCH><![CDATA[true]]></DISABLE_CASE_SENSITIVE_SEARCH>

<Data_TYPE>String List</Data_TYPE>

<DESCRIPTION><![CDATA[/home/qa desc]]></DESCRIPTION>
</SCAN_PARAMETERS>

...</CONTROL_LIST>
</RESPONSE>
</CONTROL_LIST_OUTPUT>
Compliance Policy List

/api/2.0/fo/compliance/policy/?action=list

[GET]  [POST]

View a list of compliance policies visible to the user. Policies in the XML output are sorted by compliance policy ID in ascending order. Optional input parameters support filtering the policy list output.

Maximum Policies per API Request
A maximum of 1,000 compliance policy records can be processed per request. If the requested list identifies more than 1,000 policies, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of policy records.

Permissions
Note: The Compliance APIs are available as part of one of the following subscription combinations only:
- PC and API add-on
- PC, SCA, and API add-on
- VMDR, SCA, and API add-on

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>View all compliance policies in subscription. View asset group information for all asset groups assigned to policies.</td>
</tr>
<tr>
<td>Auditor</td>
<td>View all compliance policies in subscription. View asset group information for all asset groups assigned to policies.</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>View all compliance policies in subscription. View asset group information for asset groups assigned to policies, when the user has permission to view these asset groups. This user can view groups assigned to the user’s business unit, and groups created by any user in the same business unit.</td>
</tr>
<tr>
<td>Scanner</td>
<td>View all compliance policies in subscription. View asset group information for asset groups assigned to policies, when the user has permission to view these asset groups. This user can view groups assigned to the user account, and groups created by the user.</td>
</tr>
<tr>
<td>Reader</td>
<td>View all compliance policies in subscription. View asset group information for asset groups assigned to policies, when the user has permission to view these asset groups. This user can view groups assigned to the user account, and groups created by the user.</td>
</tr>
</tbody>
</table>
**User Permissions — Asset Group Information**

Asset group information included in the policy list output includes the following, as defined for each asset group: asset group ID, title, and assigned IP addresses. Users are granted permission to view asset group information assigned to policies when the user has permission to view the asset groups.

For example, when a user makes a request for a compliance policy list and the user does not have permission to view asset groups that are assigned to the target policies, then the asset group information does not appear in the policy list output. The asset group IDs are not listed under the <POLICY> section, and the asset group title and assigned IP addresses are not listed under the <GLOSSARY> section.

In a case where a user makes a request for a compliance policy list and the user does not have permission to see one or more asset groups assigned to a target policy, the following information is provided in the compliance policy list output:

- **<POLICY> section.** The attribute “has_hidden_data=1” is returned in the <POLICY> section in the <ASSET_GROUP_IDS> element. This indicates that the user does not have permission to see one or more asset groups in the policy. When this attribute is present, only the asset group IDs that the user has permission to see, if any, are listed in the <ASSET_GROUP_IDS> element.

- **<GLOSSARY> section.** Asset group information is not displayed for asset groups assigned to compliance policies that the user does not have permission to see.

- **<WARNING_LIST> section.** A warning message is returned for informational purposes. This indicates that at least one of the compliance policies in the output has one or more asset groups that the user does not have permission to see.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>details={Basic</td>
<td>All</td>
</tr>
</tbody>
</table>
Compliance Policy List

**Sample - Compliance Policy List**

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -D headers.15 "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?action=list"
```

**XML output:**
```
<POLICY_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-11-03T21:15:29Z</DATETIME>
    <POLICY_LIST>
      <POLICY>
        <ID>18948</ID>
        <TITLE><![CDATA[XP policy]]></TITLE>
        <CREATED>
          <DATETIME>2017-10-19T18:37:15Z</DATETIME>
          <BY>quays_as</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2017-10-26T23:31:57Z</DATETIME>
          <BY>quays_as</BY>
        </LAST_MODIFIED>
        <LAST_EVALUATED>
          <DATETIME>2017-11-03T08:40:44Z</DATETIME>
        </LAST_EVALUATED>
        <STATUS><![CDATA[active]]></STATUS>
      </POLICY>
    </POLICY_LIST>
  </RESPONSE>
</POLICY_LIST_OUTPUT>
```

**DTD**
```
<platform API server>/api/2.0/fo/compliance/policy/policy_list_output.dtd
```

**Parameter** | **Description**
--- | ---
ids={value} | (Optional) Show only certain policy IDs and/or ID ranges. One or more policy IDs/ranges may be specified. Multiple entries are comma separated. A policy ID range entry is specified with a hyphen (for example, 160-165). Valid policy IDs are required.
id_min={value} | (Optional) Show only policies which have a minimum policy ID value. A valid policy ID is required.
id_max={value} | (Optional) Show only policies which have a maximum policy ID value. A valid policy ID is required.
updated_after_datetime={value} | (Optional) Show only controls updated after a certain date/time. See [Date Filters](#).
created_after_datetime={value} | (Optional) Show only controls created after a certain date/time. See [Date Filters](#).
<IS_LOCKED>0</IS_LOCKED>
<EVALUATE_NOW><![CDATA[yes]]></EVALUATE_NOW>
<ASSET_GROUP_IDS>6065</ASSET_GROUP_IDS>

<TAG_SET_INCLUDE>
  <TAG_ID>7588415</TAG_ID>
</TAG_SET_INCLUDE>

<TAG_INCLUDE_SELECTOR>ANY</TAG_INCLUDE_SELECTOR>

<ASSET_TAG_IDS>1</ASSET_TAG_IDS>

<CONTROL_LIST>
  <CONTROL>
    <ID>1045</ID>
    <STATEMENT><![CDATA[Status of the 'Clipbook' service (startup type)]]]></STATEMENT>
    <CRITICALITY>
      <LABEL><![CDATA[SERIOUS]]></LABEL>
      <VALUE>3</VALUE>
    </CRITICALITY>
  </CONTROL>

  <CONTROL>
    <ID>1048</ID>
    <STATEMENT><![CDATA[Status of the 'Shutdown: Clear virtual memory pagefile' setting]]></STATEMENT>
    <CRITICALITY>
      <LABEL><![CDATA[Critical]]></LABEL>
      <VALUE>4</VALUE>
    </CRITICALITY>
  </CONTROL>
</CONTROL_LIST>

</POLICY>
</POLICY_LIST>

<GLOSSARY>

<ASSET_GROUP_LIST>
  <ASSET_GROUP>
    <ID>6065</ID>
    <TITLE><![CDATA[Windows XP]]></TITLE>
  </ASSET_GROUP>
</ASSET_GROUP_LIST>

<ASSET_TAG_LIST>
  <TAG>
    <TAG_ID>7588415</TAG_ID>
    <TAG_NAME>Windows XP</TAG_NAME>
  </TAG>
</ASSET_TAG_LIST>
</GLOSSARY>
Compliance Policy - Export

/api/2.0/fo/compliance/policy/?action=export

[GET] [POST]

Export compliance policies from your account to an XML file. Service provided controls are exported and you can choose to also export user defined controls. The output also includes an appendix with human readable look-ups for control descriptions, giving you explanation on the various aspects of control description and evaluation.

Permissions - If you’re not a Manager, the permission to Manage PC module must be turned on in your account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=export</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>id={value} or title={value}</td>
<td>(Required) The ID or the title of the policy you want to export.</td>
</tr>
<tr>
<td>show_user_controls={0</td>
<td>1}</td>
</tr>
<tr>
<td>show_appendix={0</td>
<td>1}</td>
</tr>
<tr>
<td>show_user_controls={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

Sample - Export Policy

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
d "action=export&id=853744"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/"
```

XML output:
```
<?xml version="1.0 encoding=UTF-8" ?>
```
<DOCTYPE POLICY_EXPORT_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2/fo/compliance/policy/policy_export_output.dtd">

<POLICY>
    <TITLE><![CDATA[My Policy]]></TITLE>
    <EXPORTED><![CDATA[2013-07-17T18:19:57Z]]></EXPORTED>
    <COVER_PAGE><![CDATA[{}]]></COVER_PAGE>
    <TECHNOLOGIES total="1">
        <TECHNOLOGY>
            <ID>1</ID>
            <NAME>Windows XP desktop</NAME>
        </TECHNOLOGY>
    </TECHNOLOGIES>
    <SECTIONS total="1">
        <SECTION>
            <NUMBER>1</NUMBER>
            <HEADING><![CDATA[Default section]]></HEADING>
            <CONTROLS total="20">
                <CONTROL>
                    <ID>1111</ID>
                    <TECHNOLOGIES total="1">
                        <TECHNOLOGY>
                            <ID>1</ID>
                            <NAME>Windows XP desktop</NAME>
                            <EVALUATE checksum="74378d12a39f82721a3cb156dee58c663a650a9ce422bd311b5e5443c2af0f14"&lt;CTRL&gt;&lt;/CTRL&gt;&lt;/EVALUATE>
                        </TECHNOLOGY>
                    </TECHNOLOGIES>
                </CONTROL>
            </CONTROLS>
        </SECTION>
    </SECTIONS>
</POLICY>

Sample - Export Policy with Appendix with lookups for control descriptions

API request:

curl -u "USERNAME:PASSWORD" GET -H "X-Requested-With: curl" -X "POST" -d "action=export&id=5438&show_appendix=1" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/"showAppendix.xml
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_EXPORT_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_export_output.dtd">
<POLICY_EXPORT_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-09-09T09:07:13Z</DATETIME>
    <POLICY>
      <TITLE><![CDATA[Solaris]]></TITLE>
      <EXPORTED><![CDATA[2017-09-09T09:07:12Z]]></EXPORTED>
      <COVER_PAGE><![CDATA[]]></COVER_PAGE>
      <STATUS><![CDATA[active]]></STATUS>
      <TECHNOLOGIES total="4">
        <TECHNOLOGY>
          <ID>4</ID>
          <NAME>Solaris 9.x</NAME>
        </TECHNOLOGY>
        ...
      </TECHNOLOGIES>
    </POLICY>
  </RESPONSE>
</POLICY_EXPORT_OUTPUT>
```

Note: Remove APPENDIX section if you wish to import this XML as policy.

```xml
<OP_ACRONYMS>
  <OP id="lt">less than</OP>
  <OP id="gt">greater than</OP>
  <OP id="le">less than or equal to</OP>
  <OP id="ge">greater than or equal to</OP>
  <OP id="ne">not equal to</OP>
  <OP id="xeq">list OR string list</OP>
  <OP id="eq">equal to</OP>
  <OP id="in">in</OP>
  <OP id="xre">regular expression list</OP>
  <OP id="re">regular expression</OP>
  <OP id="range">in range</OP>
</OP_ACRONYMS>

<Data_Point_Acronyms>
  <DP>
    <!--Useraccount.legacy-plus-accounts"ự <![CDATA[The following List String value(s) <B>X</B> indicate the current list of accounts defined within the <B>/etc/group</B>, <B>/etc/shadow</B>, and/or <B>/etc/passwd</B> files having a
```
Sample - Export Library Policy to XML

You can export a library compliance policy from your account to an XML file. Just like with user created policies you must specify the input parameter show_user_controls=1 to include UDCs in the output. When the policy includes a Qualys Custom Control you’ll see the UDC ID for the control in the output.

API request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=export&ids=991742279&show_user_controls=1" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/"
```

XML output:
```
<POLICY>
  <TITLE><![CDATA[Library Policy with 2 UDC v.2.0]]></TITLE>
  <EXPORTED><![CDATA[2017-04-17T15:02:56Z]]></EXPORTED>
  <COVER_PAGE><![CDATA[]]></COVER_PAGE>
  <STATUS><![CDATA[active]]></STATUS>
  <TECHNOLOGIES total="2">
    <TECHNOLOGY>
```

```
<ID>2</ID>
  <NAME>Windows 2003 Server</NAME>
</TECHNOLOGY>
<TECHNOLOGY>
  <ID>12</ID>
  <NAME>Windows 2000</NAME>
</TECHNOLOGY>
</TECHNOLOGIES>
<SECTIONS total="1">
  <SECTION>
    <NUMBER>1</NUMBER>
    <HEADING>Untitled</HEADING>
    <CONTROLS total="1">
      <USER_DEFINED_CONTROL>
        <ID>100005</ID>
        <UDC_ID>55449d95-1877-7ee5-829a-4eedacb04f</UDC_ID>
        <CHECK_TYPE>Registry Value Existence</CHECK_TYPE>
        <IS_CONTROL_DISABLE>0</IS_CONTROL_DISABLE>
        <CATEGORY>
          <ID>3</ID>
          <NAME>Access Control Requirements</NAME>
        </CATEGORY>
        <SUBCATEGORY>
          <ID>1007</ID>
          <NAME>Authentication/Passwords</NAME>
        </SUBCATEGORY>
      </USER_DEFINED_CONTROL>
    </CONTROLS>
  </SECTION>
</SECTIONS>

Updates you’ll see once Agent UDC support is available

New Agent UDC Support will be announced soon via the Qualys Technology blog once remaining components are released.

The XML output may include the USE_AGENT_ONLY element for these Windows and Unix control types: Directory Search Control and Directory Integrity Control. This is set to 1 when the “Use agent scan only” option is enabled for the control.

The XML output may include the AUTO_UPDATE element for these Windows and Unix control types: File Integrity Control and Directory Integrity Control. This is set to 1 when the “Auto update expected value” option is enabled for the control.
Sample - Export Policy when Agent UDC Support is available

API request:

```
curl -u username:password -H "X-Requested-With: curl" -d "action=export&id=1448425&show_user_controls=1&show_appendix=0" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/" >UDCWI ND.xml
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_EXPORT_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_export_output.dtd">
<POLICY_EXPORT_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-10-05T10:41:43Z</DATETIME>
    <POLICY>
      <TITLE><![CDATA[Windows_Linux_UDC_Policy]]></TITLE>
      <EXPORTED><![CDATA[2018-10-05T10:41:43Z]]></EXPORTED>
      <COVER_PAGE></COVER_PAGE>
      <STATUS><![CDATA[active]]></STATUS>
      <TECHNOLOGIES total="3">
        <TECHNOLOGY>
          <ID>45</ID>
          <NAME>Red Hat Enterprise Linux 6.x</NAME>
        </TECHNOLOGY>
        <TECHNOLOGY>
          <ID>52</ID>
          <NAME>AIX 7.x</NAME>
        </TECHNOLOGY>
        <TECHNOLOGY>
          <ID>81</ID>
          <NAME>Red Hat Enterprise Linux 7.x</NAME>
        </TECHNOLOGY>
      </TECHNOLOGIES>
      <SECTIONS total="1">
        <SECTION>
          <NUMBER>1</NUMBER>
          <HEADING><![CDATA[ddd]]></HEADING>
          <CONTROLS total="4">
            <USER_DEFINED_CONTROL>
              <ID>100041</ID>
              <UDC_ID>929a8c4e-5057-e3f3-8225-e92d4076f499</UDC_ID>
              <CHECK_TYPE>Unix Directory Search Check</CHECK_TYPE>
            </USER_DEFINED_CONTROL>
          </CONTROLS>
        </SECTION>
      </SECTIONS>
    </POLICY>
  </RESPONSE>
</POLICY_EXPORT_OUTPUT>
```
Sample: Export Policy when Case Sensitive Search is disabled

**API Request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d "action=export&id=4034697&show_user_controls=1" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/"
```

**XML Output:**
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE POLICY_EXPORT_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_export_output.dtd">
<POLICY_EXPORT_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-07-22T08:33:50Z</DATETIME>
    <POLICY>
      <TITLE><![CDATA[Suse 11 DI and DS check]]></TITLE>
      <EXPORTED><![CDATA[2021-07-22T08:33:48Z]]></EXPORTED>
      <COVER_PAGE><![CDATA[]]></COVER_PAGE>
      <STATUS><![CDATA[active]]></STATUS>
      <TECHNOLOGIES total="2">
        <TECHNOLOGY>
          <ID>38</ID>
          <NAME>SUSE Linux Enterprise 11.x</NAME>
        </TECHNOLOGY>
      </TECHNOLOGIES>
    </POLICY>
  </RESPONSE>
</POLICY_EXPORT_OUTPUT>
```
<USER_DEFINED_CONTROL>
  <ID>100550</ID>
  <UDC_ID>74d487e1-6c1c-5de7-8063-a878ed046d7</UDC_ID>
  <CHECK_TYPE>Unix Directory Search Check</CHECK_TYPE>
  <IS_CONTROL_DISABLE><![CDATA[0]]></IS_CONTROL_DISABLE>
  <CATEGORY>
    <ID>3</ID>
    <NAME><![CDATA[Access Control Requirements]]></NAME>
  </CATEGORY>
  <SUBCATEGORY>
    <ID>1010</ID>
    <NAME><![CDATA[Account Creation/User Management]]></NAME>
  </SUBCATEGORY>
  <STATEMENT><![CDATA[Basic Directory Search Check-UNIX_edited]]></STATEMENT>
  <CRITICALITY>
    <LABEL><![CDATA[MEDIUM]]></LABEL>
    <VALUE>2</VALUE>
  </CRITICALITY>
  <COMMENT><![CDATA[Directory Search Check]]></COMMENT>
  <IGNORE_ERROR>0</IGNORE_ERROR>
  <SCAN_PARAMETERS>
    <BASE_DIR><![CDATA[/etc/123/yyy/eee/lll]]></BASE_DIR>
    <SHOULD_DESCEND><![CDATA[false]]></SHOULD_DESCEND>
    <GROUP_OWNER><![CDATA[Any Group]]></GROUP_OWNER>
    <TIME_LIMIT><![CDATA[300]]></TIME_LIMIT>
    <MATCH_LIMIT><![CDATA[50]]></MATCH_LIMIT>
    <DISABLE_CASE_SENSITIVE_SEARCH><![CDATA[false]]></DISABLE_CASE_SENSITIVE_SEARCH>
    <DATA_TYPE>String List</DATA_TYPE>
    <DESCRIPTION><![CDATA[Directory Search Check]]></DESCRIPTION>
  </SCAN_PARAMETERS>
</USER_DEFINED_CONTROL>
Database UDCs for MS SQL, Oracle, Sybase, PostgreSQL/Pivotal Greenplum, SAP IQ, and IBM DB2

You can create custom controls for MSSQL, Oracle, Sybase, PostgreSQL/Pivotal Greenplum, SAP IQ, and IBM DB2 databases. To support database controls, we’ve added new elements to the XML output and DTDs for Control List Output and Policy Export Output.

Sample - Policy API

API request:

```bash
curl -u "username:password" -H "Content-type: text/xml" -X "POST"
-d "action=export&id=1358790&show_user_controls=1"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/">
PolicyExportAPI.xml
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_EXPORT_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_
export_output.dtd">
<POLICY_EXPORT_OUTPUT>
  <RESPONSE>
    <DATETIME>2019-05-21T18:49:06Z</DATETIME>
    <POLICY>
      <TITLE>![CDATA[Objects_Check]]></TITLE>
      <EXPORTED><![CDATA[2019-05-21T18:49:06Z]]></EXPORTED>
      <COVER_PAGE></COVER_PAGE>
      <STATUS><![CDATA[active]]></STATUS>

      <USER_DEFINED_CONTROL>
        <ID>100338</ID>
        <UDC_ID>e9ff3da7-9d0c-4a64-8055-e49a3f88f838</UDC_ID>
        <CHECK_TYPE>Oracle Database Check</CHECK_TYPE>
        <IS_CONTROL_DISABLE><![CDATA[0]]></IS_CONTROL_DISABLE>
        <CATEGORY>
          <ID>5</ID>
          <NAME><![CDATA[Services]]></NAME>
        </CATEGORY>
        <SUBCATEGORY>
          <ID>1024</ID>
          <NAME><![CDATA[Support]]></NAME>
        </SUBCATEGORY>
        <STATEMENT><![CDATA[STMT:SELECT * FROM
user_tables;]]></STATEMENT>
        <CRITICALITY>
          <LABEL><![CDATA[MEDIUM]]></LABEL>
```
Sample - Export Policy for File Content Check

API request:
```bash
curl -u "username:password" -H "Content-type: text/xml" -X "POST" -d "action=export&id=1758961&show_user_controls=1" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/" FCCWin_Policy_Export.xml
```

XML output:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_EXPORT_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_export_output.dtd">
<POLICY_EXPORT_OUTPUT>
```

<RESPONSE>
  <DATETIME>2019-10-14T21:45Z</DATETIME>
  <POLICY>
    <TITLE><![CDATA[SamplePolicyWithFileContentSearchUDCs]]></TITLE>
    <EXPORTED><![CDATA[2019-10-14T21:45Z]]></EXPORTED>
    <COVER_PAGE></COVER_PAGE>
    <STATUS><![CDATA[active]]></STATUS>
    <TECHNOLOGIES total="1">
      <TECHNOLOGY>
        <ID>75</ID>
        <NAME>Windows Server 2012 R2</NAME>
      </TECHNOLOGY>
    </TECHNOLOGIES>
  </POLICY>
  <COVER_PAGE></COVER_PAGE>
  <SECTION>
    <NUMBER>1</NUMBER>
    <HEADING><![CDATA[Untitled]]></HEADING>
    <CONTROLS total="3">
      <USER_DEFINED_CONTROL>
        <ID>100006</ID>
        <UDC_ID>98e7dde1-412d-4a95-8262-b7bd168ebad8</UDC_ID>
        <CHECK_TYPE>Windows File Content Check</CHECK_TYPE>
        <IS_CONTROL_DISABLE><![CDATA[0]]></IS_CONTROL_DISABLE>
        <CATEGORY>
          <ID>8</ID>
          <NAME><![CDATA[Database Settings]]></NAME>
        </CATEGORY>
        <SUBCATEGORY>
          <ID>1044</ID>
          <NAME><![CDATA[DB Access Controls]]></NAME>
        </SUBCATEGORY>
      </USER_DEFINED_CONTROL>
      <STATEMENT><![CDATA[Windows_FCC_Use_Reg]]></STATEMENT>
      <CRITICALITY>
        <LABEL><![CDATA[min]]></LABEL>
        <VALUE>1</VALUE>
      </CRITICALITY>
      <COMMENT><![CDATA[]]></COMMENT>
      <!-- Other parameters -->
    </CONTROLS>
  </SECTION>
</RESPONSE>
**Compliance Policy - Export**

```xml
<PATH_TYPE><![CDATA[Use Registry key]]></PATH_TYPE>
<REG_HIVE><![CDATA[HKEY_CLASSES_ROOT(HKCR)]]></REG_HIVE>
<REG_KEY><![CDATA[TestKey\user]]></REG_KEY>
<REG_VALUE_NAME><![CDATA[preName]]></REG_VALUE_NAME>
<FILE_PATH><![CDATA[]]></FILE_PATH>
<FILE_QUERY><![CDATA[.*]]></FILE_QUERY>
<Data_Type>String List</Data_Type>
<DESCRIPTION><![CDATA[reg key]]></DESCRIPTION>
</SCAN_PARAMETERS>
<TECHNOLOGIES total="1">
<Technology>
<ID>75</ID>
<NAME>Windows Server 2012 R2</NAME>
<EVALUATE><CTRL><DP><K>custom.win_file_content_check.1007110</K><L>0</L><CD>contains</CD><OP>xre</OP><V><![CDATA[.*]]></V></DP></CTRL></EVALUATE>
<RATIONALE><![CDATA[rationale]]></RATIONALE>
<DATAPOINT>
<CARDINALITY>contains</CARDINALITY>
<OPERATOR>xre</OPERATOR>
<DEFAULT_VALUES total="1">
<DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
</DEFAULT_VALUES>
</DATAPOINT>
</TECHNOLOGY>
</TECHNOLOGIES>
</USER_DEFINED_CONTROL>
<USER_DEFINED_CONTROL>
<ID>100000</ID>
<UDC_ID>b24df689-0714-7045-833a-987f04cdab15</UDC_ID>
<CHECK_TYPE>Windows File Content Check</CHECK_TYPE>
<IS_CONTROL_DISABLE><![CDATA[0]]></IS_CONTROL_DISABLE>
<CATEGORY>
<ID>3</ID>
<NAME><![CDATA[Access Control Requirements]]></NAME>
</CATEGORY>
</USER_DEFINED_CONTROL>
```

740
Compliance Policy - Export

741

Management]]>

<STATEMENT><![CDATA[preFCCUDC]]></STATEMENT>

<Criticality>
  <Label><![CDATA[min]]></Label>
  <Value>1</Value>
</Criticality>

<Comment><![CDATA[]]></Comment>

<UseAgentOnly>0</UseAgentOnly>

<AutoUpdate>0</AutoUpdate>

<IgnoreError>0</IgnoreError>

<IgnoreItemNotFound>0</IgnoreItemNotFound>

<ScanParameters>
  <PathType><![CDATA[Use file search]]></PathType>
  <FileQuery><![CDATA[QWEB*]]></FileQuery>
  <BaseDir><![CDATA[c:\]]></BaseDir>
  <DepthLimit><![CDATA[3]]></DepthLimit>
  <FileNameMatch><![CDATA[preTest2.txt]]></FileNameMatch>
  <FileNameSkip><![CDATA[]]></FileNameSkip>
  <DirNameMatch><![CDATA[*]]></DirNameMatch>
  <DirNameSkip><![CDATA[]]></DirNameSkip>
  <TimeLimit><![CDATA[300]]></TimeLimit>
  <MatchLimit><![CDATA[50]]></MatchLimit>
  <DataType>String List</DataType>
  <Description><![CDATA[FileContentChech]]></Description>
</ScanParameters>

<Technologies total="1">
  <Technology>
    <Id>75</Id>
    <Name>Windows Server 2012 R2</Name>
    <Control><DP><K>custom.win_file_content_check.1007020</K><L>0</L><CD>contains</CD><OP>xre</OP><V><![CDATA[true]]></V></DP></CTRL>
  </Technology>
</Technologies>

<Rationale><![CDATA[rationale]]></Rationale>

<DataPoint>
<CARDINALITY>contains</CARDINALITY>  
<OPERATOR>xre</OPERATOR>  
<DEFAULT_VALUES total="1">  
  <DEFAULT_VALUE><![CDATA[true]]></DEFAULT_VALUE>  
</DEFAULT_VALUES>  
</DATAPoint>  
</TECHNOLOGY>  
</TECHNOLOGIES>  
</USER_DEFINED_CONTROL>  
</ID>100026</ID>  
</UDC_ID>d908b3f9-59f9-fb70-801c-29d04fb12511</UDC_ID>  
</CHECK_TYPE>Windows File Content Check</CHECK_TYPE>  
<IS_CONTROL_DISABLE><![CDATA[0]]></IS_CONTROL_DISABLE>  
</CATEGORy>  
<ID>3</ID>  
</NAME><![CDATA[Access Control Requirements]]></NAME>  
</SUBCATEGORY>  
</CATEGORY>  
</STATEMENT> <![CDATA[pre_fcc_file_path_regexwith$]]></STATEMENT>  
</CRITICALITY>  
<LABEL><![CDATA[min]]></LABEL>  
.VALUE>1</VALUE>  
</CRITICALITY>  
</COMMENT><![CDATA[]]></COMMENT>  
</USE_AGENTONLY>0</USE_AGENTONLY>  
</AUTO_UPDATE>0</AUTO_UPDATE>  
</IGNORE_ERROR>0</IGNORE_ERROR>  
</IGNORE.ITEM.NOT.FOUND>0</IGNORE.ITEM.NOT.FOUND>  
</SCAN_PARAMETERS>  
</PATH_TYPE>  
</FILE_PATH><![CDATA[C:\user\PreTest\pretestfile1.txt]]></FILE_PATH>  
   
"FILE_QUERY><![CDATA[pre$]]></FILE_QUERY>
Sample - Export policy with UDCs into XML file showing remediation information

**API request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=export&id=1801961&show_user_controls=1" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_EXPORT_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_export_output.dtd">
<POLICY_EXPORT_OUTPUT>
  <RESPONSE>
    <DATETIME>2020-04-22T16:47:24Z</DATETIME>
    
    <DATA_TYPE>String List</DATA_TYPE>
    <DESCRIPTION><![CDATA[pre]]></DESCRIPTION>
  </RESPONSE>
</POLICY_EXPORT_OUTPUT>
```
<POLICY>
  <TITLE><![CDATA[RHEL_8]]></TITLE>
  <EXPORTED><![CDATA[2020-04-22T16:47:24Z]]></EXPORTED>
  <COVER_PAGE><![CDATA[]]></COVER_PAGE>
  <STATUS><![CDATA[active]]></STATUS>
  <TECHNOLOGIES total="1">
    <TECHNOLOGY>
      <ID>217</ID>
      <NAME>Red Hat Enterprise Linux 8.x</NAME>
    </TECHNOLOGY>
  </TECHNOLOGIES>
  <SECTIONS total="2">
    ...
    <SECTION>
      <NUMBER>2</NUMBER>
      <HEADING><![CDATA[UDC]]></HEADING>
      <CONTROLS total="6">
        <USER_DEFINED_CONTROL>
          <ID>100028</ID>
          <UDC_ID>c50922a1-1482-df3f-83e2-bb96c99ff48</UDC_ID>
          <CHECK_TYPE>Unix File/Directory Permission</CHECK_TYPE>
          <IS_CONTROL_DISABLE><![CDATA[0]]></IS_CONTROL_DISABLE>
          <CATEGORY>
            <ID>3</ID>
            <NAME><![CDATA[Access Control Requirements]]></NAME>
          </CATEGORY>
          <SUB_CATEGORY>
            <ID>1007</ID>
            <NAME><![CDATA[Authentication/Passwords]]></NAME>
          </SUB_CATEGORY>
          <STATEMENT><![CDATA[Basic File/Directory Permission-RHEL_8]]></STATEMENT>
          <CRITICALITY>
            <LABEL><![CDATA[SERIOUS]]></LABEL>
            <VALUE>3</VALUE>
          </CRITICALITY>
          <COMMENT><![CDATA[Basic File/Directory Permission]]></COMMENT>
        </USER_DEFINED_CONTROL>
        <USE_AGENT_ONLY>0</USE_AGENT_ONLY>
        <AUTO_UPDATE>0</AUTO_UPDATE>
        <IGNORE_ERROR>0</IGNORE_ERROR>
        <IGNORE_ITEM_NOT_FOUND>0</IGNORE_ITEM_NOT_FOUND>
      </CONTROLS>
    </SECTION>
  </SECTIONS>
</POLICY>
<SCAN_PARAMETERS>

<FILE_PATH><![CDATA[/etc/profile]]></FILE_PATH>
<DATA_TYPE>String</DATA_TYPE>
<DESCRIPTION><![CDATA[File/Directory Permission]]></DESCRIPTION>
</SCAN_PARAMETERS>

<TECHNOLOGIES total="1">
<TECHNOLOGY>
<ID>217</ID>
<NAME>Red Hat Enterprise Linux 8.x</NAME>
<EVALUATE><CTRL><DP><K>custom.file_permission.1007079</K><OP>re</OP><V><![CDATA[.*]]></V></DP></CTRL></EVALUATE>
<RATIONALE><![CDATA[Basic File/Directory Permission-UNIX]]></RATIONALE>
<REMEDIATION><![CDATA[]]></REMEDIATION>
</TECHNOLOGY>
</TECHNOLOGIES>

<REFERENCE_LIST/>

<USER_DEFINED_CONTROL>
<ID>100029</ID>
<UDC_ID>9da2c628-fb7d-50cf-8230-6f3ff59172a8</UDC_ID>
<CHECK_TYPE>Unix File/Directory Existence</CHECK_TYPE>
<IS_CONTROL_DISABLE><![CDATA[0]]></IS_CONTROL_DISABLE>
<CATEGORY>
<ID>3</ID>
<NAME><![CDATA[Access Control Requirements]]></NAME>
</CATEGORY>
<SUB_CATEGORY>
<ID>1007</ID>
</SUB_CATEGORY>

<NAME><![CDATA[Authentication/Passwords]]></NAME>
</SUB_CATEGORY>
Compliance Policy - Export

<STATEMENT><![CDATA[Basic File/Directory Existence-UNIX-RHEL_8]]></STATEMENT>

<Criticality>
  <Label><![CDATA[SERIOUS]]></Label>
  <Value>3</Value>
</Criticality>

<Comment><![CDATA[File/Directory Existence - this is in comment section]]></Comment>

<Use_Agent_Only>0</Use_Agent_Only>

<Auto_Update>0</Auto_Update>

<Ignore_Error>0</Ignore_Error>

<Ignore_Item_Not_Found>0</Ignore_Item_Not_Found>

<Scan_Parameters>
  <File_Path><![CDATA[/etc/profile]]></File_Path>
  <Data_Type>Boolean</Data_Type>
  <Description><![CDATA[test]]></Description>
</Scan_Parameters>

<Technologies total="1">
  <Technology>
    <Id>217</Id>
    <Name>Red Hat Enterprise Linux 8.x</Name>
    <Evaluate>
      <K>custom.file_dir_exist.1007080</K><L>2</L><V>false</V>
    </Evaluate>
    <Rationale><![CDATA[File/Directory Existence-this is in rationale section under default value]]></Rationale>
    <Remediation><![CDATA[]]></Remediation>
    <DataPoint>
      <Cardinality>no cd</Cardinality>
      <Operator>no op</Operator>
      <Default_Values total="1">
        <Default_Value>true</Default_Value>
      </Default_Values>
    </DataPoint>
  </Technology>
</Technologies>

<Reference_List/>
</User_DEFINED_CONTROL>
...
</Section>
</Sections>
</Policy>
</Response>
Compliance Policy - Export

Sample: Export policy with Unix File Content Controls when Evaluate as string is enabled

API Request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d "action=export&id=3721621&show_user_controls=1" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/"

XML Output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_EXPORT_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_export_output.dtd">
<POLICY_EXPORT_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-04-06T11:56:11Z</DATETIME>
    <PROMPT>
      <TITLE><![CDATA[Multiline Check Oracle asset]]></TITLE>
      <EXPORTED><![CDATA[2021-04-06T11:56:11Z]]></EXPORTED>
      <COVER_PAGE><![CDATA[]]></COVER_PAGE>
      <STATUS><![CDATA[active]]></STATUS>
      <TECHNOLOGIES total="2">
        <TECHNOLOGY>
          <ID>79</ID>
          <NAME>Oracle Enterprise Linux 7.x</NAME>
          ...
          <CRITICALITY>
            <LABEL><![CDATA[URGENT]]></LABEL>
            <VALUE>5</VALUE>
          </CRITICALITY>
          <COMMENT><![CDATA[FC UDC]]></COMMENT>
          <USE_AGENT_ONLY>0</USE_AGENT_ONLY>
          <AUTO_UPDATE>0</AUTO_UPDATE>
          <IGNORE_ERROR>1</IGNORE_ERROR>
          <IGNORE_ITEM_NOT_FOUND>1</IGNORE_ITEM_NOT_FOUND>
          <SCAN_PARAMETERS>
            <FILE_PATH><![CDATA[/home/testscan/samram]]></FILE_PATH>
            <FILE_QUERY><![CDATA[.*]]></FILE_QUERY>
            <DATA_TYPE>Line List</DATA_TYPE>
            <EVALUATE_AS_STRING>1</EVALUATE_AS_STRING>
            <DESCRIPTION><![CDATA[New option enabled with line list]]></DESCRIPTION>
          </SCAN_PARAMETERS>
        </TECHNOLOGY>
      </TECHNOLOGIES>
    </PROMPT>
  </RESPONSE>
</POLICY_EXPORT_OUTPUT>
<TECHNOLOGIES total="2">
  <TECHNOLOGY>
    <ID>79</ID>
    <NAME>Oracle Enterprise Linux 7.x</NAME>
  </TECHNOLOGY>
  ...
</TECHNOLOGIES>

**DTD**

<platform API server>/api/2/fo/compliance/policy/policy_export_output.dtd
Compliance Policy - Import

/api/2.0/fo/compliance/policy/?action=import

[POST]

Import a compliance policy, defined in an XML file, into your account. We’ll include all the service-provided controls from your XML file. You have the option to also include user-defined controls.

Permissions - If you’re not a Manager, the permission to Manage PC module must be turned on in your account.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=import</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=0[1]</td>
<td>(Optional) Show (echo) the request’s input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.</td>
</tr>
<tr>
<td>xml_file</td>
<td>(Required) The file containing the policy details.</td>
</tr>
<tr>
<td>title={value}</td>
<td>(Required) The title of the new policy.</td>
</tr>
<tr>
<td>create_user_controls=0[1]</td>
<td>(Optional) When not specified, user-defined controls are not created when you import a policy. Specify 1 to include UDCs from the XML file.</td>
</tr>
</tbody>
</table>

Sample - Import policy

API request:

```
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2017-09-15T21:32:40Z</DATETIME>
    <TEXT>Successfully imported compliance policy</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Sample - Import policy with UDCs having remediation information using xml file

API request:

```bash
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2020-04-22T22:51:16Z</DATETIME>
    <TEXT>Successfully imported compliance policy</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>1867541</VALUE>
      </ITEM>
      <ITEM>
        <KEY>TITLE</KEY>
        <VALUE>Policy1</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
**Compliance Policy - Merge**

/api/2.0/fo/compliance/policy/?action=merge

[POST]

Merge (combine) 2 or more compliance policies using Qualys Policy Compliance (PC). You can choose to merge some or all parts of a new policy into an existing one. Also you can preview merge changes before saving them. This API is available to Managers and Auditors.

For example, say you imported a policy from our library (Policy A) and configured it to add asset groups, controls and sections. Later we might release an updated version of this policy (Policy B) with new controls and technologies. In this scenario you can use the Policy Merge API to add the new controls and technologies from Policy B into Policy A (your existing policy) without losing the asset groups, controls and sections you added.

**Input Parameters**

The policy merge input parameters give you flexibility with merging different parts of a new policy (Policy B) into an existing one (Policy A). For example you can choose to update controls with newer definitions, replace asset groups, and add new technologies and controls. By default no changes are applied to your existing policy unless parameters are specified (see below).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=merge</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) The ID of the policy that will be updated with merged content (let's call this Policy A).</td>
</tr>
<tr>
<td>merge_policy_id={value}</td>
<td>(Required) Tell us the policy with the content that will be merged into Policy A (let's call this Policy B). You can specify a policy ID using &quot;merge_policy_id&quot; or policy XML data. To upload XML data, use this syntax: --data-binary @path_to_xml_file.xml</td>
</tr>
<tr>
<td>replace_cover_page={0</td>
<td>1}</td>
</tr>
<tr>
<td>replace_asset_groups={0</td>
<td>1}</td>
</tr>
<tr>
<td>add_asset_groups={0</td>
<td>1}</td>
</tr>
<tr>
<td>add_new_technologies={0</td>
<td>1}</td>
</tr>
</tbody>
</table>
Policy Merge Request 1 - preview merged policy

Policy ID 15993 (Policy A) will be updated with content merged from policy ID 15994 (Policy B) and the XML output will show the merged policy in preview mode. Policy changes will not be saved in Policy 15993 since the request includes "preview_merge=1".

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?action=merge&id=15993&merge_policy_id=15994&replace_cover_page=1&add_new_asset_groups=1&add_new_technologies=1&update_section_heading=1&add_new_controls=1&update_existing_controls=1&preview_merge=1"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_MERGE_RESULT_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_merge_result_output.dtd">
<POLICY_MERGE_RESULT_OUTPUT>
RESPONSE>
<DATETIME>2018-04-24T05:28:04Z</DATETIME>
<POLICY_MERGE_RESULT>
<NOTE>Policy changes were not merged or saved since the request had preview_merge=1.</NOTE>
<NEW_COVER_PAGE><![CDATA[My Cover Page]]></NEW_COVER_PAGE>
<ASSET_GROUPS_ADDED>
<ASSET_GROUP>
,ID>424422</ID>
<NAME><![CDATA[<script>alert("xss");]]></NAME>></ASSET_GROUP>
</ASSET_GROUPS_ADDED>
</POLICY_MERGE_RESULT>
</RESPONSE>
</POLICY_MERGE_RESULT_OUTPUT>

Parameter Description

add_new_controls={0|1} (Optional) Set add_new_controls=1 to add new controls, i.e. add controls from Policy B if they are not already in Policy A.

update_section_heading={0|1} (Optional) Set update_section_heading=1 to replace the section heading in Policy A with the one in Policy B, based on section number (applies only to common sections).
This parameter must be specified with: add_new_controls or update_existing_controls.

update_existing_controls={0|1} (Optional) Set update_existing_controls=1 to replace the common controls in Policy A with the ones in Policy B. These are controls that exist in both policies. (Controls will not be removed).

preview_merge={0|1} (Optional) Set preview_merge= 1 to view the changes merged into Policy A without saving them.

DTD

<platform API server>/api/2.0/fo/compliance/policy/policy_merge_result_output.dtd"
Compliance Policy - Merge

</ASSET_GROUP>
<ASSET_GROUP>
  <ID>424577</ID>
  <NAME><![CDATA[10.10.32.26]]></NAME>
</ASSET_GROUP>
</ASSET_GROUPS_ADDED>
<TECHNOLOGIES_ADDED>
  <TECHNOLOGY>
    <ID>1</ID>
    <NAME>Windows XP desktop</NAME>
  </TECHNOLOGY>
</TECHNOLOGIES_ADDED>
<SECTIONS_UPDATED>
  <SECTION>
    <ID>1</ID>
    <HEADING><![CDATA[First section]]></HEADING>
  </SECTION>
  <SECTION>
    <ID>2</ID>
    <HEADING><![CDATA[Second section]]></HEADING>
  </SECTION>
</SECTIONS_UPDATED>
<SECTIONS>
  <SECTION>
    <ID>1</ID>
    <CONTROLS_UPDATED>
      <CONTROL>
        <ID>1061</ID>
      </CONTROL>
    </CONTROLS_UPDATED>
  </SECTION>
  <SECTION>
    <ID>2</ID>
    <CONTROLS_ADDED>
      <CONTROL>
        <ID>1045</ID>
      </CONTROL>
      <CONTROL>
        <ID>1048</ID>
      </CONTROL>
    </CONTROLS_ADDED>
  </SECTION>
</SECTIONS>
</POLICY_MERGE_RESULT>
</RESPONSE>
Policy Merge Request 2 - save merged policy

Policy ID 15993 (Policy A) will be updated with content merged from policy ID 15994 (Policy B). The merged policy will be saved in policy 15993.

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?
action=merge&id=15993&merge_policy_id=15994&replace_cover_page=1&add_new_asset_groups=1&add_new_technologies=1&update_section_heading=1&add_new_controls=1&update_existing_controls=1&preview_merge=0"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_MERGE_RESULT_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_merge_result_output.dtd">
<POLICY_MERGE_RESULT_OUTPUT>
<RESPONSE>
 <DATETIME>2018-04-24T05:31:26Z</DATETIME>
 <POLICY_MERGE_RESULT>
  <NOTE>Policy changes have been merged successfully.</NOTE>
  <NEW_COVER_PAGE><![CDATA[My Cover Page]]></NEW_COVER_PAGE>
  <ASSET_GROUPS_ADDED>
   <ASSET_GROUP>
    <ID>424422</ID>
   </ASSET_GROUP>
  </ASSET_GROUPS_ADDED>
 </POLICY_MERGE_RESULT>
</RESPONSE>
</POLICY_MERGE_RESULT_OUTPUT>
```

Policy Merge Request 3 - pass policy XML, preview merged policy

Policy ID 15993 (Policy A) will be updated with content merged from the policy defined in the file "path_to_policy_xml_file.xml." The merged changes will not be saved in policy 15993 since the request includes “preview_merge=1”.

API request:
```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -H
"Content-type: text/xml"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?
action=merge&id=15993&replace_cover_page=1&replace_asset_groups=1&add_new_technologies=1&update_section_heading=1&add_new_controls=1&update_existing_controls=1&preview_merge=1" --data-binary
@/home/aamin/PC_XML/path_to_policy_xml_file.xml>
```
XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_MERGE_RESULT_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_merge_result_output.dtd">
<POLICY_MERGE_RESULT_OUTPUT>
  <RESPONSE>
    <DATETIME>2018-04-24T05:38:26Z</DATETIME>
    <POLICY_MERGE_RESULT>
      <NOTE>Policy changes were not merged or saved since the request had preview_merge=1.</NOTE>
      <NEW_COVER_PAGE><![CDATA[My Cover Page]]></NEW_COVER_PAGE>
      <SECTIONS_UPDATED>
        <SECTION>
          <ID>1</ID>
          <HEADING><![CDATA[First section]]></HEADING>
        </SECTION>
        <SECTION>
          <ID>2</ID>
          <HEADING><![CDATA[Second section]]></HEADING>
        </SECTION>
      </SECTIONS_UPDATED>
      <SECTIONS>
        <SECTION>
          <ID>1</ID>
          <CONTROLS_UPDATED>
            <CONTROL>
              <ID>1061</ID>
            </CONTROL>
          </CONTROLS_UPDATED>
        </SECTION>
        <SECTION>
          <ID>2</ID>
          <CONTROLS_ADDED>
            <CONTROL>
              <ID>1045</ID>
            </CONTROL>
            <CONTROL>
              <ID>1048</ID>
            </CONTROL>
          </CONTROLS_ADDED>
        </SECTION>
      </SECTIONS>
    </POLICY_MERGE_RESULT>
  </RESPONSE>
</POLICY_MERGE_RESULT_OUTPUT>
```
</RESPONSE>
</POLICY_MERGE_RESULT_OUTPUT>
Compliance Policy - Manage Asset Tags

/api/2.0/fo/compliance/policy/

[POST]

Add, remove, and set asset tags for a policy. You must have permission to modify the policy you want to update.

Add asset tags to policy

Use this action to add asset tags to the policy. When specified, we will check whether the asset tags specified in the request are already associated with the policy and only add the asset tags that are new to the policy.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=add_asset_tags</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) Policy ID for the policy you want to update.</td>
</tr>
<tr>
<td>evaluate_now={0</td>
<td>1}</td>
</tr>
<tr>
<td>tag_include_selector={all</td>
<td>any}</td>
</tr>
<tr>
<td>tag_exclude_selector={all</td>
<td>any}</td>
</tr>
<tr>
<td>tag_set_by={id</td>
<td>name}</td>
</tr>
<tr>
<td>tag_set_include={tag id</td>
<td>name}</td>
</tr>
<tr>
<td>tag_set_exclude={tag id</td>
<td>name}</td>
</tr>
</tbody>
</table>

API request:

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "id=4201701&tag_set_include=118766028&tag_include_selector=all" https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?action=add_asset_tags"

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2022-01-19T06:35:40Z</DATETIME>
    <TEXT>Compliance Policy successfully modified.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>4201701</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>

Removing asset tags from policy

Use this action to remove asset tags from the policy.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=remove_asset_tags</td>
<td>(Required) Note: With the remove_asset_tags action, you must set either</td>
</tr>
<tr>
<td></td>
<td>tag_set_include or tag_set_exclude parameter, or both the parameters.</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) Policy ID for the policy you want to update.</td>
</tr>
<tr>
<td>evaluate_now={0</td>
<td>1}</td>
</tr>
<tr>
<td></td>
<td>assigned assets, and select the Evaluate Now check box in the UI Policy</td>
</tr>
<tr>
<td></td>
<td>Editor. When this check box is selected we’ll start policy evaluation each</td>
</tr>
<tr>
<td></td>
<td>time you save changes to the policy from the UI or API.</td>
</tr>
<tr>
<td>tag_include_selector={all</td>
<td></td>
</tr>
<tr>
<td>any}</td>
<td>one of the selected tags. Select “all” to include hosts that match all of</td>
</tr>
<tr>
<td></td>
<td>the selected tags.</td>
</tr>
<tr>
<td>tag_exclude_selector={all</td>
<td></td>
</tr>
<tr>
<td>any}</td>
<td>one of the selected tags. Select “all” to exclude hosts that match all of</td>
</tr>
<tr>
<td></td>
<td>the selected tags.</td>
</tr>
<tr>
<td>tag_set_by={id</td>
<td>name}</td>
</tr>
<tr>
<td></td>
<td>IDs. Specify “name” to select a tag set by providing tag names.</td>
</tr>
<tr>
<td>tag_set_include={tag id</td>
<td>name}</td>
</tr>
<tr>
<td></td>
<td>be included. You identify the tag set by providing tag name or IDs. Multiple</td>
</tr>
<tr>
<td></td>
<td>entries are comma separated.</td>
</tr>
<tr>
<td>tag_set_exclude={tag id</td>
<td>name}</td>
</tr>
<tr>
<td></td>
<td>be excluded. You identify the tag set by providing tag name or IDs. Multiple</td>
</tr>
<tr>
<td></td>
<td>entries are comma separated.</td>
</tr>
</tbody>
</table>
API Request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "id=4201701&tag_set_include=118766028&tag_include_selector=all" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?action=remove_asset_tags"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2022-01-19T06:35:40Z</DATETIME>
    <TEXT>Compliance Policy successfully modified.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>4201701</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Set asset tags for policy
Use this action to overwrite the asset tags for a policy. Any assigned asset tags not specified in the request will be removed from the policy.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=set_asset_tags</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>(Required) Policy ID for the policy you want to update.</td>
</tr>
<tr>
<td>evaluate_now={0</td>
<td>1}</td>
</tr>
<tr>
<td>tag_include_selector={all</td>
<td>any}</td>
</tr>
<tr>
<td>tag_exclude_selector={all</td>
<td>any}</td>
</tr>
<tr>
<td>tag_set_by={id</td>
<td>name}</td>
</tr>
</tbody>
</table>
Compliance Policy - Manage Asset Tags

Parameter | Description
--- | ---
tag_set_include={tag id|name} | (Optional) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
tag_set_exclude={tag id|name} | (Optional) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.

**API Request:**
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "id=4201701&tag_set_include=118766028&tag_include_selector=all" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?action=set_asset_tags"
```

**XML output:**
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2022-01-19T06:28:30Z</DATETIME>
    <TEXT>Compliance Policy successfully modified.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>4201701</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Compliance Policy - Manage Asset Groups

@api/2.0/fo/compliance/policy/

[POST]

Add, remove and set asset groups for a policy. You must have permission to modify the policy you want to update.

Add asset group IDs to policy

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=add_asset_group_ids</td>
<td>(Required) Policy ID for the policy you want to update.</td>
</tr>
<tr>
<td>id={value}</td>
<td>Policy ID for the policy you want to update.</td>
</tr>
<tr>
<td>asset_group_ids={value}</td>
<td>Asset groups IDs for the asset groups you want to add to the policy specified in “id”. Multiple IDs are comma separated. Each asset group must have at least 1 assigned IP address.</td>
</tr>
<tr>
<td>evaluate_now={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -X POST -d "id=43400&asset_group_ids=649737,649736" "https://qualysapi.qualys.com//api/2.0/fo/compliance/policy/?action=add_asset_group_ids"
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-09-11T09:06:17Z</DATETIME>
    <TEXT>Compliance Policy successfully modified.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>43400</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Remove asset group IDs from policy

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=remove_asset_group_ids</td>
<td>(Required)</td>
</tr>
<tr>
<td>id={value}</td>
<td>Policy ID for the policy you want to update.</td>
</tr>
<tr>
<td>asset_group_ids={value}</td>
<td>Asset groups IDs for the asset groups you want to delete from the policy specified in “id”. Multiple IDs are comma separated.</td>
</tr>
<tr>
<td>evaluate_now={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

API request:
```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -X POST -d "id=43400&asset_group_ids=649737,649736" 
"https://qualysapi.qualys.com//api/2.0/fo/compliance/policy/?action=remove_asset_group_ids"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM 
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
 <RESPONSE>
  <DATETIME>2014-09-11T09:06:17Z</DATETIME>
  <TEXT>Compliance Policy successfully modified.</TEXT>
  <ITEM_LIST>
   <ITEM>
    <KEY>ID</KEY>
    <VALUE>43400</VALUE>
   </ITEM>
  </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Set asset group IDs for policy

Use this action to reset the asset groups for a specified policy. Any assigned asset groups not specified in this request will be removed.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=set_asset_group_ids</td>
<td>(Required) Policy ID for the policy you want to update.</td>
</tr>
<tr>
<td>id={value}</td>
<td>Asset groups IDs for the asset groups you want to assign to the policy specified in &quot;id&quot;. Multiple IDs are comma separated. Each asset group must have at least 1 assigned IP address.</td>
</tr>
<tr>
<td>asset_group_ids={value}</td>
<td>(Optional) Specify evaluate_now=1 to immediately evaluate the policy against assigned assets, and select the Evaluate Now check box in the UI Policy Editor. When this check box is selected we'll start policy evaluation each time you save changes to the policy from the UI or API.</td>
</tr>
<tr>
<td>evaluate_now={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

API request:
```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -X POST -d "id=43400&asset_group_ids=649737,649736" "https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?action=set_asset_group_ids"
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-09-11T09:07:43Z</DATETIME>
    <TEXT>Compliance Policy successfully modified.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>43400</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```
Compliance Posture Information

The Policy Compliance APIs help you gain the essential insight into the compliance posture of the hosts within your account. Qualys PC provides you with the following two types of APIs to fetch posture data:

- PC Posture Information APIs
- PC Posture Streaming APIs

Depending on the size of your posture data, you may choose to use the PC Posture Information APIs or the PC Posture Streaming APIs.

We recommend that you leverage the Posture Streaming APIs if you want to download more than 1 million postures, as they provide you a faster way to retrieve larger amount of posture information. The Posture Streaming APIs also has the ability to filter data based on the desired input entered by the user. If your posture information is expected to be less than 1 million, use the PC Posture Information APIs.

Refer the following KB article for recommendations on retrieving compliance posture information:

Compliance Posture Data Retrieval Best Practices

PC Posture Information APIs

/api/2.0/fo/compliance/posture/info/?action=list

[GET] [POST]

Each compliance posture info record includes a compliance posture ID and other attributes. Optional input parameters support filtering the posture info record output.

Each compliance posture info record in the output includes:

<table>
<thead>
<tr>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compliance Posture ID</td>
<td>The service assigns a unique value to each compliance posture info record.</td>
</tr>
<tr>
<td>Host ID</td>
<td>Identifies a host.</td>
</tr>
<tr>
<td>Control ID</td>
<td>Identifies a technical control.</td>
</tr>
<tr>
<td>Technology ID</td>
<td>Identifies a technology.</td>
</tr>
<tr>
<td>Instance</td>
<td>Identifies a technology instance, when applicable.</td>
</tr>
<tr>
<td>Compliance Status</td>
<td>Passed, Failed or Error. An error, only assigned to a custom control, indicates control evaluation failed (and the ignore errors configuration option for the control was not selected).</td>
</tr>
<tr>
<td>Evaluation Date</td>
<td>The last posture evaluation date.</td>
</tr>
<tr>
<td>First Fail Date</td>
<td>The first scan date when the control was reported as Fail. If the previous status was Pass then this is the date the status changed from Pass to Fail.</td>
</tr>
<tr>
<td>Last Fail Date</td>
<td>The most recent scan date when the control was reported as Fail.</td>
</tr>
</tbody>
</table>
The user has the ability to select the amount of information to include in the posture information output. By default, basic posture information is included: the posture ID, host ID, control ID, technology ID, technology instance (when applicable), and the compliance status. If an exception has been created, this full exception information is also included: the exception assignee and status, the date/time when the exception was created, when it was last modified, the user who took these actions on the exception, and the date when the exception is set to expire. A glossary of compliance posture information identifies: basic host information and basic control information.

Use the details input parameter to select another level of detail to be included in the policy information output.

By default, the posture information output shows posture information for all hosts (IP addresses) in asset groups assigned to the selected policy, provided the user has permission to view the hosts themselves. If you have a sub-account like a Unit Manager, Scanner or Reader, the posture information output only includes hosts that the account has permission to see.

**Best Practices**

You can reduce the amount of data being retrieved by only pulling the data that is required for the downstream processes. For example, only download the delta of the changes in posture since the last pull. This can be done using optional input parameters which allow you to set filters to restrict the posture information output to postures info records with certain IP addresses, host IDs, compliance control IDs, compliance posture IDs, posture info records with changes in status since a specified date, and posture info records with a certain compliance status (Passed, Failed or Error).
The optional glossary in the compliance posture information output includes:

<table>
<thead>
<tr>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User List</td>
<td>List of users who created, modified, or added comments to exceptions in compliance posture info records which are included in the posture information output. For a policy that was edited, the user who most recently edited the exception is listed.</td>
</tr>
<tr>
<td>Host List</td>
<td>List of hosts in compliance posture info records which are included in the posture information output. This basic host information is included: host ID, IP address, and tracking method. When details=All is specified, this additional information is included: last vulnerability scan date/time, last compliance scan date/time.</td>
</tr>
<tr>
<td>Control List</td>
<td>List of controls in compliance posture info records which are included in the posture information output. When details=All is specified, this additional information is included: rationale information and technology information for each control.</td>
</tr>
<tr>
<td>Technology List</td>
<td>List of technologies for controls in compliance posture info records which are included in the posture list output. This information is included only when details=All is specified.</td>
</tr>
<tr>
<td>Evidence List</td>
<td>List of evidence information for control data points.</td>
</tr>
</tbody>
</table>

**Maximum Postures per API Request**

The output of the Compliance Posture Info API is paginated when your API request identifies a single policy to report on using the “policy_id” input parameter. In this case, a maximum of 5,000 posture info records are returned per request by default. You can customize the page size (i.e. the number of posture info records) by using the parameter “truncation_limit=10000” for instance if you want to return pages with 10,000 records.

**Permissions**

Note: The Posture Info API is available as part of one of the following subscription combinations only:

- PC and API add-on
- PC, SCA, and API add-on
- VMDR, SCA, and API add-on

All users have permission view posture information for hosts (IP addresses) in asset groups assigned to the selected policy, when the hosts are available to the user based on user account settings.

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>View compliance postures for all hosts (IP addresses) in asset groups assigned to the selected policy.</td>
</tr>
<tr>
<td>Auditor</td>
<td>View compliance postures for all hosts (IP addresses) in asset groups assigned to the selected policy.</td>
</tr>
</tbody>
</table>
User Permissions: Asset Group IPs

All users have permission to view posture information for all hosts (IP addresses) in the asset groups assigned to the selected policy provided they have permission to view the hosts themselves. This permission is granted even when users do not have permission to view the asset groups assigned to the policy.

For example, when a user makes a request for compliance posture information for “Policy A” and this policy has one assigned asset group “Hong Kong”, and the user does not have permission to view this asset group, then the user does have permission to view compliance posture info records for all the IP addresses in the asset group “Hong Kong” provided the IP addresses in the group “Hong Kong” are visible to the user.
## Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>policy_id={value}</td>
<td>(policy_id or policy_ids is required) Show compliance posture info records for a specified policy. A valid policy ID is required. The parameters policy_id and policy_ids cannot be specified in the same request.</td>
</tr>
<tr>
<td>policy_ids={value}</td>
<td>(policy_id or policy_ids is required) Show compliance posture info records for multiple policies - up to 10 policies may be requested. Provide a comma-separated list of valid policy IDs. When this parameter is specified, all posture data is downloaded (and the “truncation_limit” parameter is invalid). The parameters policy_id and policy_ids cannot be specified in the same request. When policy_ids is specified, truncation_limit is invalid. For CSV output, policy_id must be specified (and policy_ids is invalid).</td>
</tr>
<tr>
<td>echo_request={0</td>
<td>1}</td>
</tr>
<tr>
<td>output_format={value}</td>
<td>(Optional) The output format. A valid value is: xml (default), csv (posture data and metadata, i.e. summary and warning data), csv_no_metadata (posture data only, no metadata). For CSV output, you can include only one policy. For this reason, policy_id is required.</td>
</tr>
</tbody>
</table>
| details={Basic|All|None|Light} | (Optional) Show a certain amount of information for each compliance posture info record. A valid value is:  

- None - show posture info and minimum exception information (assignee and status) if appropriate  
- Basic (default) - show posture info, full exception information if appropriate, and a minimum glossary (basic info for hosts and controls)  
- Light - show posture info, exception info if appropriate, and a limited glossary (host info and last scan date/time, control ID, and evidence info)  
- All - show posture info (including the percentage of controls that passed for each host), exception info if appropriate, posture summary (the number of assets, controls, and control instances evaluated) and a glossary (host info and last scan date/time), control info, technology info, evidence info  

When hide_evidence=1 is specified in the same request as details=All or details=Light, then evidence info will not be shown in the output. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hide_evidence={0</td>
<td>1}</td>
</tr>
<tr>
<td>show_extended_evidence={0</td>
<td>1}</td>
</tr>
<tr>
<td>include_dp_name={value}</td>
<td>(Optional) Show the name and ID for each data point in the XML output. This is useful for uniquely identifying data points.</td>
</tr>
<tr>
<td>show_remediation_info={0</td>
<td>1}</td>
</tr>
<tr>
<td>cause_of_failure={0</td>
<td>1}</td>
</tr>
<tr>
<td>truncation_limit={value}</td>
<td>(Optional) The parameter is valid only when the API request is for a single policy and the policy_id parameter is specified.</td>
</tr>
<tr>
<td></td>
<td>By default, a limit of 5,000 posture info records are returned per request (when &quot;policy_id&quot; is specified). You may specify a value less than the default (1-4999) or greater than the default (5001-1000000) to configure the number records returned per request.</td>
</tr>
<tr>
<td></td>
<td>If the requested list identifies more records than the truncation limit, then the XML output includes the &lt;WARNING&gt; element and the URL for making another request for the next batch of records.</td>
</tr>
<tr>
<td></td>
<td>You can specify truncation_limit=0 for no truncation limit. This means that the output is not paginated and all the records are returned in a single output. WARNING: This can generate very large output and processing large XML files can consume a lot of resources on the client side. In this case it is recommended to use the pagination logic and parallel processing. The previous page can be processed while the next page is being downloaded.</td>
</tr>
<tr>
<td>ips={value}</td>
<td>(Optional) Show only compliance posture info records for compliance hosts which have certain IP addresses/ranges. One or more IP addresses/ranges may be specified. Multiple IPs/ranges are comma separated.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>host_ids={value}</td>
<td>(Optional) Show only compliance posture info records for compliance hosts which have certain host IDs and/or ID ranges. One or more host IDs/ranges may be specified. Multiple entries are comma separated. A host ID range entry is specified with a hyphen (for example, 123-125). Valid host IDs are required.</td>
</tr>
<tr>
<td>control_ids={value}</td>
<td>(Optional) Show only compliance posture info records for controls which have certain control IDs and/or ranges. One or more control IDs/ranges may be specified. Multiple entries are comma separated. An control ID range entry is specified with a hyphen (for example, 1200-1300). Valid control IDs are required.</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Optional) Show only compliance posture info records for certain compliance posture IDs and/or ID ranges. One or more posture IDs/ranges may be specified. Multiple entries are comma separated. A posture ID range entry is specified with a hyphen (for example, 1-10). Valid posture IDs are required.</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(Optional) Show only compliance posture info records which have a minimum ID value. A valid posture ID is required.</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(Optional) Show only compliance posture info records which have a maximum ID value. A valid posture ID is required.</td>
</tr>
<tr>
<td>status_changes_since=</td>
<td>(Optional) Show compliance posture info records when the compliance status was changed since a certain date and time (optional). If the policy itself was changed, a warning message is generated. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2008-05-01” or “2008-05-01T23:12:00Z”.</td>
</tr>
<tr>
<td>evaluation_date={date}</td>
<td>(Optional) Show compliance posture info records when the posture evaluation date is equal to or greater than a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2021-04-01” or “2021-04-01T23:12:00Z”.</td>
</tr>
<tr>
<td>asset_group_ids={value}</td>
<td>(Optional) Show only hosts in certain asset groups. Provide a comma-separated list of asset group IDs for the asset groups you want to download compliance posture data for. The asset groups specified do not need to be assigned to the one or more policies requested. Posture data will be returned as long as there are common hosts specified by “asset_group_ids” and asset groups that are assigned to the policies requested.</td>
</tr>
<tr>
<td>status={Passed</td>
<td>Failed</td>
</tr>
</tbody>
</table>
### Sample - Posture Info filtered by evaluation date

In this example, we’re filtering the output by an evaluation date of 2021-03-05. The XML output will only include info records with an evaluation date equal to or greater than March 5, 2021.

**API request:**
Compliance
Compliance Posture Information

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d "action=list&policy_id=3318470&details=Basic&output_format=xml&evaluation_date=2021-03-05" "https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/

XML Response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POSTURE_INFO_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/posture_info_list_output.dtd">
<POSTURE_INFO_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-04-07T21:59:40Z</DATETIME>
    <INFO_LIST>
      <INFO>
        <ID>10911451</ID>
        <HOST_ID>3077710</HOST_ID>
        <CONTROL_ID>1071</CONTROL_ID>
        <TECHNOLOGY_ID>43</TECHNOLOGY_ID>
        <INSTANCE></INSTANCE>
        <STATUS>Passed</STATUS>
        <POSTURE_MODIFIED_DATE>2020-11-03T07:12:32Z</POSTURE_MODIFIED_DATE>
        <EVALUATION_DATE>2021-04-05T20:36:21Z</EVALUATION_DATE>
        <PREVIOUS_STATUS>Passed</PREVIOUS_STATUS>
        <FIRST_FAIL_DATE>N/A</FIRST_FAIL_DATE>
        <LAST_FAIL_DATE>N/A</LAST_FAIL_DATE>
        <FIRST_PASS_DATE>2020-11-03T07:12:32Z</FIRST_PASS_DATE>
        <LAST_PASS_DATE>2021-04-05T20:36:22Z</LAST_PASS_DATE>
      </INFO>
      <INFO>
        <ID>10911452</ID>
        <HOST_ID>3077710</HOST_ID>
        <CONTROL_ID>1113</CONTROL_ID>
        <TECHNOLOGY_ID>43</TECHNOLOGY_ID>
        <INSTANCE></INSTANCE>
        <STATUS>Failed</STATUS>
        <POSTURE_MODIFIED_DATE>2020-11-03T07:12:32Z</POSTURE_MODIFIED_DATE>
        <EVALUATION_DATE>2021-04-05T20:36:21Z</EVALUATION_DATE>
        <PREVIOUS_STATUS>Failed</PREVIOUS_STATUS>
        <FIRST_FAIL_DATE>2020-11-03T07:12:32Z</FIRST_FAIL_DATE>
        <LAST_FAIL_DATE>2021-04-05T20:36:22Z</LAST_FAIL_DATE>
        <FIRST_PASS_DATE>N/A</FIRST_PASS_DATE>
        <LAST_PASS_DATE>N/A</LAST_PASS_DATE>
      </INFO>
    </INFO_LIST>
  </RESPONSE>
</POSTURE_INFO_LIST_OUTPUT>
```
Sample - Posture Info with Data Point Name

Sample API request to uniquely identify Data Points using Name and ID.

**API request:**
```
curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWORD" -d headers.15 'https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/?action=list&policy_id=15472&details=All&include_dp_name=1'
```

**XML Response:**
```
...```
Compliance Posture Information

Sample - Posture Info with database controls
This applies to database UDCs for Oracle, MSSQL, Sybase, PostgreSQL/Pivotal Greenplum, SAP IQ, and IBM DB2.

When the Posture API output includes database controls, the values returned for the database controls are shown in a tabular format. You’ll see these elements in the output: Header (H), Row (R) and Column (C).

API request:
```
curl -u "username:password" -H "Content-type: text/xml" -X "POST"
-d "action=list&policy_id=1303776&details=All&include_dp_name=1"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/"
PostureInfo.xml
```

XML output:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POSTURE_INFO_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/posture_info_list_output.dtd">
<POSTURE_INFO_LIST_OUTPUT>
<RESPONSE>
<DATETIME>2019-05-03T19:24:32Z</DATETIME>
/INFO_LIST>
/INFO
```

<DPV_LIST>
    <DPV lastUpdated="2019-05-03T00:33:14Z">
        <LABEL>:dp_2</LABEL>
        <V>
            <H>
                <C><![CDATA[CustomerID]]></C>
                <C><![CDATA[CustomerName]]></C>
                <C><![CDATA[ContactName]]></C>
                <C><![CDATA[Address]]></C>
                <C><![CDATA[City]]></C>
                <C><![CDATA[PostalCode]]></C>
                <C><![CDATA[Country]]></C>
            </H>
            <R>
                <C><![CDATA[1]]></C>
                <C><![CDATA[Alfreds Futterkiste]]></C>
                <C><![CDATA[Maria Anders]]></C>
                <C><![CDATA[Obere Str. 57]]></C>
                <C><![CDATA[Berlin]]></C>
                <C><![CDATA[12209]]></C>
                <C><![CDATA[Germany]]></C>
            </R>
            <R>
                <C><![CDATA[2]]></C>
                <C><![CDATA[Ana Trujillo Emparedados y helados]]></C>
                <C><![CDATA[Ana Trujillo]]></C>
                <C><![CDATA[Avda. de la Constitucion 2222]]></C>
                <C><![CDATA[Mexico D.F.]]></C>
                <C><![CDATA[05021]]></C>
                <C><![CDATA[Mexico]]></C>
            </R>
            <R>
                <C><![CDATA[3]]></C>
                <C><![CDATA[Antonio Moreno Taqueria]]></C>
                <C><![CDATA[Antonio Moreno]]></C>
                <C><![CDATA[Mataderos 2312]]></C>
                <C><![CDATA[Mexico D.F.]]></C>
                <C><![CDATA[05023]]></C>
                <C><![CDATA[Mexico]]></C>
            </R>
        </R>
    </DPV>
</DPV_LIST>

...
Sample - Posture Info for File Content Check

API request:

curl -u "username:password" -H "Content-type: text/xml" -X "POST" -d "action=list&echo_request=1&policy_id=1758961&details=All&include_dp_name=1" https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/">

posture_info_result.xml

XML output:

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POSTURE_INFO_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/posture_info_list_output.dtd">
<POSTURE_INFO_LIST_OUTPUT>
<REQUEST>
<DATETIME>2019-10-14T21:19:57Z</DATETIME>
<User_LOGIN>rey_pt11</User_LOGIN>

<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/</RESOURCE>

<PARAM_LIST>
<PARAM>
<KEY>action</KEY>
<VALUE>list</VALUE>
</PARAM>
<PARAM>
<KEY>echo_request</KEY>
<VALUE>1</VALUE>
</PARAM>
<PARAM>
<KEY>policy_id</KEY>
<VALUE>1758961</VALUE>
</PARAM>
<PARAM>
<KEY>details</KEY>
<VALUE>All</VALUE>
</PARAM>
<PARAM>
<KEY>include_dp_name</KEY>
<VALUE>1</VALUE>
</PARAM>
</PARAM_LIST>
</REQUEST>

<RESPONSE>
<DATETIME>2019-10-14T21:19:57Z</DATETIME>
<INFO_LIST>
  <INFO>
    <ID>34544283</ID>
    <HOST_ID>7368441</HOST_ID>
    <CONTROL_ID>100006</CONTROL_ID>
    <TECHNOLOGY_ID>75</TECHNOLOGY_ID>
    <INSTANCE></INSTANCE>
    <STATUS>Passed</STATUS>
    <POSTURE_MODIFIED_DATE>2019-10-14T21:15:46Z</POSTURE_MODIFIED_DATE>
    <EVALUATION_DATE>2019-10-14T21:15:46Z</EVALUATION_DATE>
    <PREVIOUS_STATUS>Passed</PREVIOUS_STATUS>
    <FIRST_FAIL_DATE>N/A</FIRST_FAIL_DATE>
    <LAST_FAIL_DATE>N/A</LAST_FAIL_DATE>
    <FIRST_PASS_DATE>2019-10-14T21:15:46Z</FIRST_PASS_DATE>
    <LAST_PASS_DATE>2019-10-14T21:15:46Z</LAST_PASS_DATE>
    <EVIDENCE>
      <BOOLEAN_EXPR><![CDATA[:dp_2 contains $tp_2]]></BOOLEAN_EXPR>
      <DPV_LIST>
        <DPV lastUpdated="2019-10-14T19:53:41Z">
          <LABEL>:dp_2</LABEL>
          <V fileName="c:\Agent\user\test2.txt"><![CDATA[QWEB]]></V>
          <TM_REF>@tm_1</TM_REF>
        </DPV>
      </DPV_LIST>
    </EVIDENCE>
  </INFO>
  <INFO>
    <ID>34544284</ID>
    <HOST_ID>7368441</HOST_ID>
    <CONTROL_ID>100000</CONTROL_ID>
    <TECHNOLOGY_ID>75</TECHNOLOGY_ID>
    <INSTANCE></INSTANCE>
    <STATUS>Failed</STATUS>
    <POSTURE_MODIFIED_DATE>2019-10-14T21:15:46Z</POSTURE_MODIFIED_DATE>
    <EVALUATION_DATE>2019-10-14T21:15:46Z</EVALUATION_DATE>
    <PREVIOUS_STATUS>Failed</PREVIOUS_STATUS>
    <FIRST_FAIL_DATE>2019-10-14T21:15:46Z</FIRST_FAIL_DATE>
    <LAST_FAIL_DATE>2019-10-14T21:15:46Z</LAST_FAIL_DATE>
    <FIRST_PASS_DATE>N/A</FIRST_PASS_DATE>
    <LAST_PASS_DATE>N/A</LAST_PASS_DATE>
    <EVIDENCE>
      <BOOLEAN_EXPR><![CDATA[:dp_1 contains $tp_1]]></BOOLEAN_EXPR>
  </INFO>
</INFO_LIST>
<DPV_LIST>
  <DPV lastUpdated="2019-10-14T19:53:41Z">
    <LABEL>:dp_1</LABEL>
    <V fileName="C:\preTest2.txt"><![CDATA[QWEB]]></V>
    <TM_REF>@tm_2</TM_REF>
  </DPV>
</DPV_LIST>
</EVIDENCE>
</INFO>
<INFO>
  <ID>34544285</ID>
  <HOST_ID>7368441</HOST_ID>
  <CONTROL_ID>100026</CONTROL_ID>
  <TECHNOLOGY_ID>75</TECHNOLOGY_ID>
  <INSTANCE></INSTANCE>
  <STATUS>Passed</STATUS>
  <POSTURE_MODIFIED_DATE>2019-10-14T21:15:46Z</POSTURE_MODIFIED_DATE>
  <EVALUATION_DATE>2019-10-14T21:15:46Z</EVALUATION_DATE>
  <PREVIOUS_STATUS>Passed</PREVIOUS_STATUS>
  <FIRST_FAIL_DATE>N/A</FIRST_FAIL_DATE>
  <LAST_FAIL_DATE>N/A</LAST_FAIL_DATE>
  <FIRST_PASS_DATE>2019-10-14T21:15:46Z</FIRST_PASS_DATE>
  <LAST_PASS_DATE>2019-10-14T21:15:46Z</LAST_PASS_DATE>
  <EVIDENCE>
    <BOOLEAN_EXPR><![CDATA[:dp_3 contains $tp_2]]></BOOLEAN_EXPR>
  </EVIDENCE>
</INFO>
<INFO_LIST>
<SUMMARY>
  <TOTAL_ASSETS>1</TOTAL_ASSETS>
  <TOTAL_CONTROLS>3</TOTAL_CONTROLS>
  <CONTROL_INSTANCES>
    <TOTAL>3</TOTAL>
    <TOTAL_PASSED>2</TOTAL_PASSED>
    <TOTAL_FAILED>1</TOTAL_FAILED>
    <TOTAL_ERROR>0</TOTAL_ERROR>
  </CONTROL_INSTANCES>
</SUMMARY>
<TOTAL_EXCEPTIONS>0</TOTAL_EXCEPTIONS>
</CONTROL_INSTANCES>
</SUMMARY>
<GLOSSARY>
<HOST_LIST>
<HOST>
<ID>7368441</ID>
<IP>10.115.74.93</IP>
<TRACKING_METHOD>AGENT</TRACKING_METHOD>
<DNS><![CDATA[win-890blrmesc6]]></DNS>
<NETBIOS><![CDATA[WIN-890BLRMESC6]]></NETBIOS>
<OS><![CDATA[Windows Server 2012 R2 Standard 64 bit Edition]]></OS>
<QG_HOSTID>3031a534-6b78-4c4c-aacd-db56257c155f</QG_HOSTID>
<ASSET_ID>689027</ASSET_ID>
<LAST_VULN_SCAN_DATETIME>2019-10-14T19:18:12Z</LAST_VULN_SCAN_DATETIME>
<LAST_COMPLIANCE_SCAN_DATETIME>2019-10-14T20:21:07Z</LAST_COMPLIANCE_SCAN_DATETIME>
<PERCENTAGE><![CDATA[66.67% (2 of 3)]]></PERCENTAGE>
</HOST>
</HOST_LIST>
<CONTROL_LIST>
<CONTROL>
<ID>100006</ID>
<STATEMENT><![CDATA[Windows_FCC_Use_Reg]]></STATEMENT>
<CRITICALITY>
<LABEL><![CDATA[min]]></LABEL>
.VALUE>1</VALUE>
</CRITICALITY>
<RATIONALE_LIST>
<RATIONALE>
<TECHNOLOGY_ID>75</TECHNOLOGY_ID>
<TEXT><![CDATA[rationale]]></TEXT>
</RATIONALE>
</RATIONALE_LIST>
</CONTROL>
<CONTROL>
>ID>100000</ID>
<STATEMENT><![CDATA[preFCCUDC]]></STATEMENT>
<CRITICALITY>
<LABEL><![CDATA[min]]></LABEL>
.VALUE>1</VALUE>
</CRITICALITY>
<RATIONALE_LIST>
<RATIONALE>
<TECHNOLOGY_ID>75</TECHNOLOGY_ID>
<TEXT><![CDATA[rationale]]></TEXT>
</RATIONALE>
</RATIONALE_LIST>
</CONTROL>
<CONTROL>
<ID>100026</ID>

<STATEMENT><![CDATA[pre_fcc_file_path_regexwith$]]></STATEMENT>
<CRITICALITY>
<LABEL><![CDATA[min]]></LABEL>
(VALUE>1</VALUE>
</CRITICALITY>
<RATIONALE_LIST>
<RATIONALE>
<TECHNOLOGY_ID>75</TECHNOLOGY_ID>
<TEXT><![CDATA[ration]]></TEXT>
</RATIONALE>
</RATIONALE_LIST>
</CONTROL>
</CONTROL_LIST>
<TECHNOLOGY_LIST>
<TECHNOLOGY>
<ID>75</ID>
<NAME><![CDATA[Windows Server 2012 R2]]></NAME>
</TECHNOLOGY>
</TECHNOLOGY_LIST>
<DPD_LIST>
<DPD>
<LABEL>:dp_1</LABEL>
<ID>1007020</ID>

<NAME><![CDATA[custom.win_file_content_check.1007020]]></NAME>
<DESC><![CDATA[FileContentChech]]></DESC>
</DPD>

<DPD>
<LABEL>:dp_2</LABEL>
>ID>1007110</ID>

<NAME><![CDATA[custom.win_file_content_check.1007110]]></NAME>
<DESC><![CDATA[reg key]]></DESC>
</DPD>

<DPD>
<LABEL>:dp_3</LABEL>
<ID>1008003</ID>

</DPD>
</DPD_LIST>
<NAME><![CDATA[custom.win_file_content_check.1008003]]></NAME>
  <DESC><![CDATA[pre\$]]></DESC>
</DPD>
</DPD_LIST>
<TP_LIST>
<TP>
  <LABEL>$tp_1</LABEL>
  <V><![CDATA[true]]></V>
</TP>
<TP>
  <LABEL>$tp_2</LABEL>
  <V><![CDATA[.*]]></V>
</TP>
</TP_LIST>
<TM_LIST>
<TM>
  <LABEL>@tm_1</LABEL>
  <PAIR>
    <K><![CDATA[item not found:2]]></K>
    <V><![CDATA[Set status Passed for 'item not found\error']></V>
  </PAIR>
</TM>
<TM>
  <LABEL>@tm_2</LABEL>
  <PAIR>
    <K><![CDATA[item not found:2]]></K>
    <V><![CDATA[Set status Passed for 'item not found\error']></V>
  </PAIR>
</TM>
<TM>
  <LABEL>@tm_3</LABEL>
  <PAIR>
    <K><![CDATA[item not found:2]]></K>
    <V><![CDATA[Set status Passed for 'item not found\error']></V>
  </PAIR>
</TM>
</TM_LIST>
</GLOSSARY>
</RESPONSE>
</POSTURE_INFO_LIST_OUTPUT>
Sample - Posture Info with Extended Evidence parameter

This sample includes 2 INFO records. One record has data for Extended Evidence, and the other record has data for Statistics and Extended Statistics Error.

API request:

```bash
curl -H "X-Requested-With:curl" -u "USERNAME:PASSWORD" -d "action=list&policy_id=1055704&details=All&output_format=xml&show_extended_evidence=1" "http://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/
```

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POSTURE_INFO_LIST_OUTPUT SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/posture_info_list_output.dtd">
<POSTURE_INFO_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2022-04-28T11:08:05Z</DATETIME>
    <POLICY>
      <ID>1055704</ID>
      <DATETIME>2022-04-28T11:08:05Z</DATETIME>
    </POLICY>
    <INFO_LIST>
      <INFO>
        <ID>10461454</ID>
        <HOST_ID>2573671</HOST_ID>
        <CONTROL_ID>1071</CONTROL_ID>
        <TECHNOLOGY_ID>80</TECHNOLOGY_ID>
        <INSTANCE><![CDATA[os]]></ INSTANCE>
        <STATUS>Failed</STATUS>
        <POSTURE_MODIFIED_DATE>2022-04-27T08:57:38Z</POSTURE_MODIFIED_DATE>
        <EVALUATION_DATE>2022-05-02T06:21:45Z</EVALUATION_DATE>
        <PREVIOUS_STATUS>Failed</PREVIOUS_STATUS>
        <FIRST_FAIL_DATE>2022-04-27T08:57:38Z</FIRST_FAIL_DATE>
        <LAST_FAIL_DATE>2022-05-02T06:21:45Z</LAST_FAIL_DATE>
        <FIRST_PASS_DATE>2022-04-21T12:05:56Z</FIRST_PASS_DATE>
        <LAST_PASS_DATE>2022-04-21T12:05:56Z</LAST_PASS_DATE>
        <EVIDENCE>
          <BOOLEAN_EXPR><![CDATA[(:dp_2 in #fv_2 or :dp_2 < $tp_1)]]></ BOOLEAN_EXPR>
          <DPV_LIST>
            <DPV lastUpdated="2022-04-28T10:03:33Z">
              <LABEL>:dp_2</LABEL>
              <V><![CDATA[5]]></ V>
            </DPV>
          </DPV_LIST>
          <EXTENDED_EVIDENCE><![CDATA[Row 1:File name,Setting,Value Row 2:/etc/login.defs,PASS_MIN_LEN,5]]></ EXTENDED_EVIDENCE>
          <STATISTICS><![CDATA[]]></STATISTICS>
          <EXTENDED_STATISTICS_ERROR><![CDATA[]]></EXTENDED_STATISTICS_ERROR>
        </EVIDENCE>
      </INFO>
    </INFO_LIST>
  </RESPONSE>
</POSTURE_INFO_LIST_OUTPUT>
```
Compliance

Compliance Posture Information

<INFO>

<ID>10479751</ID>

<HOST_ID>2573673</HOST_ID>

<CONTROL_ID>100002</CONTROL_ID>

<TECHNOLOGY_ID>81</TECHNOLOGY_ID>

<INSTANCE><![CDATA[os]]></INSTANCE>

<STATUS>Passed</STATUS>

<POSTURE_MODIFIED_DATE>2022-04-28T10:09:39Z</POSTURE_MODIFIED_DATE>

<EVALUATION_DATE>2022-05-02T06:21:45Z</EVALUATION_DATE>

<PREVIOUS_STATUS>Passed</PREVIOUS_STATUS>

<FIRST_FAIL_DATE>N/A</FIRST_FAIL_DATE>

<LAST_FAIL_DATE>N/A</LAST_FAIL_DATE>

<FIRST_PASS_DATE>2022-04-28T10:09:39Z</FIRST_PASS_DATE>

<LAST_PASS_DATE>2022-05-02T06:21:45Z</LAST_PASS_DATE>

<BOOLEAN_EXPR><![CDATA[ :dp_8 matches $tp_5 ]]></BOOLEAN_EXPR>

<DPV_LIST>

<DPV lastUpdated="2022-04-28T10:03:26Z">

<label> :dp_8 </label>

<![CDATA[ No data found ]]

</DPV>

</DPV_LIST>

<STATISTICS><![CDATA[ Search duration: 63 seconds ]]

<EXTENDED_STATISTICS_ERROR><![CDATA[ Error Code 28: Base directory not found can't lstat target of '/usr/lib/debug/usr/.dwz' (No such file or directory), can't lstat target of '/usr/lib/systemd/system/dbs-org.freedesktop.network1.service' (No such file or directory), can't lstat target of '/usr/src/kernels/3.10.0-327.el7.x86_64/build' (No such file or directory), can't lstat target of '/usr/src/kernels/3.10.0-327.el7.x86_64/source' (No such file or directory), can't lstat target of '/usr/share/gdb/auto-load/bin' (No such file or directory), can't lstat target of '/usr/share/gdb/auto-load/lib' (No such file or directory), can't lstat target of '/usr/share/gdb/auto-load/sbin' (No such file or directory), can't lstat target of '/usr/share/PkgKit/icons' (No such file or directory) ]]

</EXTENDED_STATISTICS_ERROR>

</INFO>
PC Posture Streaming APIs

The PC Posture Streaming APIs help you have posture data continuously streamed for one host ID at a time for all the specified host IDs for which posture information is available. You can process the data concurrently, which helps you save time and achieve optimum results within a short span of time.

You must use the compliance posture streaming APIs in the following sequence:

- **Get Policy List (optional):** Use the Get Policy List API to fetch the list of policies against which you want to evaluate the compliance posture of your assets.
  
  If you have a list of policy IDs, you can directly use the Resolve Host IDs API.

- **Resolve Host IDs:** Use the policy IDs received from Get Policy List API as the input parameter for Resolve Host IDs. You can specify a maximum of 10 comma-separated policy IDs.

- **Get Posture Info:** Use the host IDs received from Resolve Host IDs as the input parameter to get the posture info in JSON stream.

Note: By default, for each of these APIs, you can send a maximum of 25 API requests per Qualys PC subscription in a 60-second timeframe.

To customize the rate limits as per your requirement, contact your Technical Account Manager or Qualys support.

Before you start, you must authenticate to receive the token needed for the PC Posture API requests. The API Gateway is responsible for authenticating the user who accesses the Posture Streaming APIs to get the posture details.

Input parameters for Auth API:

- username
- password
- token

URL for Auth API: https://gateway.<assigned URL>/auth

```python
headers = {'Content-Type': 'application/x-www-form-urlencoded'}
authUrl = 'https://gateway.<assigned URL>/auth'
data = {'username': 'username',
        'password': 'password',
        'token': 'true'}
authResp = requests.post(authUrl, data=data, headers=headers, verify=False)
token = authResp.content.decode('utf-8')
```

**Note:** Provide the URL, user ID, and password that have been assigned to you. For token, the value can be ‘true’ or ‘True’.
Compliance Posture Information

Get Policy List

API URL:

/pcreds/1.0/posture/policy/list?lastEvaluationDate=<evaluation date>

[GET]
List all policy IDs as per the last evaluation date specified.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization</td>
<td>(Required) JWT encrypted token.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Provide the token received from the Authorization API as the input.</td>
</tr>
<tr>
<td>lastEvaluationDate</td>
<td>(Optional) Compliance posture information records when the posture is equal to or greater than the specified date. You may also specify the time. The format for date and time is YYYY-MM-DD or YYYY-MM-DDTHH:MM:SSZ (UTC/GMT).</td>
</tr>
</tbody>
</table>

Sample JSON output response - Get Policy List

Get Policy List without lastEvaluationDate

Request:

curl -X GET https://gateway.<assigned URL>/pcrs/1.0/posture/policy/list -H "accept: */*
-H "Authorization: Bearer <token>"

Response:

```json
{
    "subscriptionId": "<SUBSCRIPTION ID>",
    "policyList": [
        {
            "id": "<POLICY ID>",
            "title": "VMWARE 5.5 AND 6.0",
            "createdBy": "<USER NAME>",
            "createdDate": "2018-01-19T07:52:11Z",
            "modifiedBy": "<USER NAME>",
            "modifiedDate": "2018-02-01T09:32:33Z",
            "lastEvaluatedDate": "2019-12-19T06:38:46Z",
            "status": "inactive",
            "locked": 0
        },
        {
```

785
Get Policy List with lastEvaluationDate

Request:


Response:

{
    "subscriptionId": "<SUBSCRIPTION ID>",
    "policyList": [
        {
            "id": "<POLICY ID>",
            "title": "ALL",
            "createdBy": "<USER NAME>",
            "createdDate": "2021-04-08T05:46:39Z",
        }
    ]
}
Resolve Host IDs

API URL:
/pcrs/1.0/posture/hostids?policyId=policyId1, policyId2

[GET]

List all Host IDs for the specified policies.

For the Resolve Host IDs API, you must use the token that is returned by the authentication request. You can include a maximum of 10 policies in one API request.
Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policyId</td>
<td>(Required) Policy IDs for compliance evaluation.</td>
</tr>
<tr>
<td>Request header:</td>
<td>Authorization (Required) JWT encrypted token.</td>
</tr>
<tr>
<td></td>
<td>Note: Provide the token received from the Authorization API as the input.</td>
</tr>
<tr>
<td>lastScanDate</td>
<td>(Optional) Get host IDs based on the specified date on which they were last scanned.</td>
</tr>
<tr>
<td></td>
<td>The formats for date are:</td>
</tr>
<tr>
<td></td>
<td>lastScanDate=YYYY-MM-DD</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>lastScanDate=2022-05-25</td>
</tr>
<tr>
<td></td>
<td>Or,</td>
</tr>
<tr>
<td></td>
<td>lastScanDate=2022-05-25T18:48:16Z</td>
</tr>
</tbody>
</table>

Sample JSON output response - Resolve Host IDs

Resolve Host IDs with single policy ID

Request:


Response:

[  
  {  
    "policyId": "<POLICY ID>",  
    "subscriptionId": "SUBSCRIPTION ID",  
    "hostIds": [  
      "<HOST ID>"  
    ]  
  }  
]
Resolve Host IDs with multiple policy IDs

Request:

Response:
[

{
    "policyId": "<POLICY ID>",
    "subscriptionId": "<SUBSCRIPTION ID>",
    "hostIds": [
        "<HOST ID>
    ]
},

{
    "policyId": "<POLICY ID>",
    "subscriptionId": "SUBSCRIPTION ID",
    "hostIds": [
        "<HOST ID>
    ]
}
]

Resolve Host IDs based on the specified date

Request:
curl -X GET https://gateway.<assigned URL>/pcrs/1.0/posture/hostids?policyId=4677689&lastScanDate=2022-05-25 -H "accept: */*" -H "Authorization: <token>"

Response:
[

{
    "policyId": "4677689",
]
Get Posture Info

API URL:

/pcrs/1.0/posture/postureInfo?evidenceRequired=0&compressionRequired=1&lastEvaluationDate=2021-12-23

[POST]

Get continuous posture information for all the specified hosts for each policy ID included in the API.

To get posture information, you must use the host IDs retrieved in the Resolve Host IDs API request.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>evidenceRequired</td>
<td>Default value is 0, which indicates that evidence data will not be retrieved for the host posture. If you want evidence data to be retrieved, change the value to 1. <strong>Note:</strong> Changing the value to 1 will increase the time required to fetch posture data.</td>
</tr>
<tr>
<td>compressionRequired</td>
<td>Default value is 1, which indicates that the output will be compressed. If you do not want the data to be compressed, change the value to 0. <strong>Note:</strong> Not compressing the data will increase the time required to fetch posture data.</td>
</tr>
</tbody>
</table>

Request Body: Output of the Resolve Host ID and the JWT token.

Request header: Authorization [Required] JWT encrypted token. **Note:** The token received from the Authorization API and the token used in the second API need to be the input here.
### Compliance Posture Information

#### Sample JSON output response - Get Posture Info (single policy ID)

Get Posture Info with lastEvaluationDate, without evidence, without compression, without lastScanDate

**Request:**

```bash
curl -X POST "https://gateway.<assigned URL>/pcrs/1.0/posture/postureInfo?evidenceRequired=0&compressionRequired=0&lastEvaluationDate=2021-12-23" -H "accept: */*" -H "Authorization: Bearer <token>" -H "Content-Type: application/json" -d "{"policyId":"<POLICY ID>"","subscriptionId":"<SUBSCRIPTION "}
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lastEvaluationDate</td>
<td>(Optional) Compliance posture information records when the posture is equal to or greater than the specified date. You may also specify the time. The formats for date and time are: YYYY-MM-DD Or, YYYY-MM-DDTHH:MM:SSZ (UTC/GMT).</td>
</tr>
<tr>
<td>lastScanDate</td>
<td>(Optional) Compliance posture information on the date on which an asset was last scanned. The formats for date are: lastScanDate=2021-12-17 Or, lastScanDate=2021-12-17T18:48:16Z</td>
</tr>
<tr>
<td>lastScanDateFrom, lastScanDateTo</td>
<td>(Optional) Compliance posture information of the assets scanned between these two dates, both dates included. The format for dates is: lastScanDateFrom=2022-09-30 or 2022-09-30T18:48:16Z lastScanDateTo=2022-12-27 or 2022-12-27T20:48:16Z</td>
</tr>
<tr>
<td>statusChangedSince</td>
<td>(Optional) Compliance posture information records when the posture is changed in policy since the specified date. You may also specify the time. The formats for date and time are: YYYY-MM-DD Or, YYYY-MM-DDTHH:MM:SSZ (UTC/GMT)</td>
</tr>
</tbody>
</table>

**Notes:**
- You must specify both dates.
- You must not use these parameters with lastScanDate.
Response:

```json
{
  "id": xxx,
  "instance": "os",
  "policyId": "<POLICY ID>",
  "policyTitle": "<POLICY TITLE>",
  "netBios": "<NETBIOS>",
  "controlId": "<CONTROL ID>",
  "controlStatement": "Status of the 'banner motd' configuration command on the device",
  "rationale": "The 'Message of the Day (banner motd)' command is used to provide a warning banner displayed when a connection to the device is made BEFORE a user successfully authenticates to the device. The Message of the Day banner can be used to provide an acceptable use policy or warning prior to login notifying that all user activity may be monitored and potential legal consequences may result from unauthorized use. Run this check periodically to ensure content of the banner displayed is in compliance with the requirements and expectations driven by internal standards and/or policies."
    
  "remediation": "Execute following commands to set desired banner message:
1. configure terminal
2. banner motd 'delimiting-character' 'message' 'delimiting-character'
3. exit

    
  "controlReference": null,
  "technologyId": xxx,
  "status": "Error",
  "previousStatus": "Error",
  "firstFailDate": "",
  "lastFailDate": "",
  "firstPassDate": "2021-12-21T11:28:21Z",
  "lastPassDate": "2021-12-21T11:29:22Z",
  "postureModifiedDate": "2021-12-22T12:56:41Z",
  "lastEvaluatedDate": "2021-12-23T05:32:40Z",
  "created": "2022-02-21T13:10:13Z",
  "hostId": "<HOST ID>",
  "ip": "xx.xx.xx.xxx",
  "trackingMethod": "IP",
  "os": null,
  "osCpe": "cpe:/o:cisco::7.0%283%29i2%282%29:::",
  "domainName": "<DOMAIN NAME>",
  "dns": null,
  "qgHostid": null,
  "networkId": 0,
  "networkName": "Global Default Network",
  "complianceLastScanDate": "2021-12-22T12:49:59Z",
  "customerUuid": "<CUSTOMER UUID>",
```

792.
"customerId": "<CUSTOMER ID>",
"assetId": "<ASSET ID>",
"technology": {
   "id": xxx,
   "name": "Cisco NX-OS"
},
"criticality": {
   "label": "CRITICAL",
   "value": 4
},
"evidence": null,
"causeOfFailure": null,
"currentBatch": 1,
"totalBatches": 1
}

Get Posture Info without lastEvaluationDate, without evidence, without compression, without lastScanDate

User input: evidenceRequired=0 and compressionRequired=0

Request:

{
"policyId": "<POLICY ID>",
"policyTitle": "<POLICY TITLE>",
"netBios": "<NETBIOS>",
"controlId": "<CONTROL ID>",
"technologyId": "<TECHNOLOGY ID>",
"status": "Failed",
"previousStatus": "Failed",
"firstFailDate": "2021-10-25T07:21:13Z",
"lastFailDate": "2021-10-29T07:52:41Z",
"firstPassDate": "",
"lastPassDate": "",
"postureModifiedDate": "2021-10-25T07:21:11Z",
"lastEvaluatedDate": "2021-10-29T07:52:41Z",
"created": "2021-10-29T07:54:26Z",
}"
]"
"hostId": "<HOST ID>",
"ip": "xx.xx.xx.xx",
"trackingMethod": "IP",
"os": null,
"osCpe": "cpe:/o:microsoft:windows_server_2012:r2::x64:",
"domainName": "<DOMAIN NAME>",
"dns": "comdevsql2016",
"qgHostid": null,
"networkId": "0",
"networkName": "Global Default Network",
"complianceLastScanDate": "2021-10-28T16:57:58Z",
"customerUuid": "xxx",
"customerId": "xxx",
"assetId": "xxx",
"technology": {
  "id": xx,
  "name": "Windows Server 2012 R2"
},
"criticality": {
  "label": "SERIOUS",
  "value": 3
},
"evidence": null,
"causeOfFailure": {
  "missing": {
    "logic": null,
    "value": [
      1,
      "Attribute not found",
      "Unable to retrieve password policy"
    ]
  },
  "unexpected": {
    "value": [0]
  }
}
}
{
  "id": xxx,
  "instance": "os",
  "policyId": "<POLICY ID>",
  "policyTitle": "<POLICY TITLE>",
  "netBios": "<NETBIOS>"
Get Posture Info without lastEvaluationDate, without evidence, with compression, without lastScanDate

User input: evidenceRequired=0 & compressionRequired=1

Request:
curl -X POST "https://gateway.<assigned URL>/pcrs/1.0/posture/postureInfo?evidenceRequired=0&compressionRequired=1"
Get Posture Info without lastEvaluationDate, with evidence, with compression, without lastScanDate

User input: evidenceRequired=1 & compressionRequired=1

Request:
Get Posture Info without lastEvaluationDate, with evidence, without compression, without lastScanDate

User input: evidenceRequired=1, compressionRequired=0

Request:
```
"policyId": "xxx",
"instance": "os",
"policyId": <POLICY ID>,
"policyTitle": <POLICY TITLE>,
"netBios": <NETBIOS>,
"controlId": <CONTROL ID>,
"technologyId": <TECHNOLOGY ID>,
"status": "Failed",
"previousStatus": "Failed",
"firstFailDate": "2021-10-25T07:13Z",
"lastFailDate": "2021-10-29T07:52:41Z",
"firstPassDate": "]
```

Response:
```
[{
  "id": xxx,
  "instance": "os",
  "policyId": <POLICY ID>,
  "policyTitle": <POLICY TITLE>,
  "netBios": <NETBIOS>,
  "controlId": <CONTROL ID>,
  "technologyId": <TECHNOLOGY ID>,
  "status": "Failed",
  "previousStatus": "Failed",
  "firstFailDate": "2021-10-25T07:13Z",
  "lastFailDate": "2021-10-29T07:52:41Z",
  "firstPassDate": "]
```
"lastPassDate": "",
"postureModifiedDate": "2021-10-25T07:21:11Z",
"lastEvaluatedDate": "2021-10-25T07:52:41Z",
"created": "2021-10-29T07:55:26Z",
"hostId": <HOST ID>,
"ip": "xx.xx.xx.xx",
"trackingMethod": "IP",
"os": null,
"osCpe":
  "cpe:/o:microsoft:windows_server_2012:r2::x64:",
"domainName": "<DOMAIN NAME>",
"dns": "comdevsql2016",
"qgHostId": null,
"networkId": "0",
"networkName": "Global Default Network",
"complianceLastScanDate": "2021-10-28T16:39:55Z",
"customerUuid": "0a387e70-8b26-78ff-8145-017b816fa17f",
"customerId": "<CUSTOMER ID>",
"assetId": "<ASSET ID>",
"technology": {
  "id": xx,
  "name": "Windows Server 2012 R2"
},
"criticality": {
  "label": "SERIOUS",
  "value": 3
},
"evidence": {
  "expectedValues": "\nAttribute not found\n---------
OR ------------
Unable to retrieve password policy\n------ OR ------------
equal to
1",
  "currentValues": [
    "0"
  ],
  "actualValues": null,
  "directoryFimUdc": null
},
"causeOfFailure": {
  "missing": {
    "logic": null,
    "value": [
      "Attribute not found",
      "Unable to retrieve password policy"
    ]
  },
  "value": [
    "1",
    "Attribute not found",
  ]
}
"unexpected": { 
  "value": [ 
    "0"
  ]
}
],
{
  "id": xx,
  "instance": "MSSQL 2016:1:1433:MSSQLSERVER:PCDEV",
  "policyId": "<POLICY ID>",
  "policyTitle": "<POLICY TITLE>",
  "netBios": "<NETBIOS>",
  "controlId": "<CONTROL ID>",
  "technologyId": "<TECHNOLOGY ID>",
  "status": "Passed",
  "previousStatus": "Passed",
  "firstFailDate": "",
  "lastFailDate": "",
  "firstPassDate": "2021-10-25T07:21:13Z",
  "lastPassDate": "2021-10-29T07:52:41Z",
  "postureModifiedDate": "2021-10-25T07:21:11Z",
  "lastEvaluatedDate": "2021-10-29T07:52:41Z",
  "created": "2021-10-29T07:55:27Z",
  "hostId": "<HOST ID>",
  "ip": "xx.xx.xx.xx",
  "trackingMethod": "IP",
  "os": null,
  "osCpe": "cpe:/o:microsoft:windows_server_2012:r2::x64:",
  "domainName": "<DOMAIN NAME>",
  "dns": "comdevsq12016",
  "qgHostId": null,
  "networkId": "0",
  "networkName": "Global Default Network",
  "complianceLastScanDate": "2021-10-28T16:39:55Z",
  "customerUuid": "<CUSTOMER UUID>",
  "customerId": "<CUSTOMER ID>",
  "assetId": "<ASSET ID>",
  "technology": { 
    "id": xx,
    "name": "Microsoft SQL Server 2016"
  },
  "criticality": { 
    "label": "SERIOUS",
    "value": 3
Compliance Posture Information

Sample JSON output response - Get Posture Info (multiple policy IDs)

Get Posture Info with lastEvaluationDate, with evidence, without compression, without lastScanDate

User input: evidenceRequired=1 & compressionRequired=0

Request:

curl -X POST "https://gateway.<assigned URL>/pcrs/1.0/posture/postureInfo?evidenceRequired=1&compressionRequired=0&lastEvaluationDate=2021-12-27T15:35:22Z" -H "accept: /" -H "Authorization: Bearer <token>" -H "Content-Type: application/json" -d "[{{"policyId":"<Policy_ID>","subscriptionId":"<Subscription_ID>","hostIds":[]}],{{"policyId":"<policyId1>","subscriptionId":"<Subscription_ID>","hostIds":[]}}]"

Response:

[
  {
    "id": xx,
    "instance": "os",
    "policyId": <POLICY_ID>,
    "policyTitle": "<POLICY TITLE>",
    "netBios": "<NETBIOS>",
    "controlId": <CONTROL_ID>,
    "controlStatement": "Status of the 'Minimum Password Length' setting",
    "rationale": "Among the several characteristics that make 'user identification' via password a secure and workable solution is setting a 'minimum password length' requirement. Each character that is added to the password length squares the difficulty of breaking the password via 'brute force,' which attempts using every combination possible within the password symbol set-space, in order

800
to discover a user's password. While no 'minimum length' can be
 guaranteed secure, eight (8) is commonly considered to be the
 minimum for most application access, along with requiring other
 password security factors, such as increasing the size of the
 symbol set-space by requiring mixed-cases, along with other forms
 of password variability creation, increases the difficulty of
 breaking any password by brute-force attack.

"remediation": "To establish the recommended configuration
via GP, set the following UI path to 14 or more
character(s):\n\n\nComputer Configuration\Policies\Windows
Settings\Security Settings\Account Policies\Password
Policy\Minimum password length",
"controlReference": null,
"technologyId": xx,
"status": "Passed",
"previousStatus": "Passed",
"firstFailDate": "",
"lastFailDate": "",
"firstPassDate": "2021-10-12T13:12:26Z",
"lastPassDate": "2021-12-27T15:35:22Z",
"postureModifiedDate": "2021-10-12T13:12:26Z",
"lastEvaluatedDate": "2021-12-27T15:35:22Z",
"created": "2022-02-24T14:21:06Z",
"hostId": xx,
"ip": "xx.xx.xx.xx",
"trackingMethod": "DNS Hostname",
"os": xx,
"osCpe": "cpe:/o:microsoft:windows_2003_server::sp2::",
"domainName": "<DOMAIN NAME>",
"dns": "client5-25-244.root.vuln.qa.qualys.com",
"qgHostid": xx,
"networkId": xx,
"networkName": "Global Default Network",
"complianceLastScanDate": "2021-12-27T15:31:18Z",
"customerUuid": "xx",
"customerId": "xx",
"assetId": xx,
"technology": {
"id": xx,
"name": "Windows 2003 Server"
},
"criticality": {
"label": "CRITICAL",
"value": xx
},
"evidence": {
"expectedValues": "\ngreater than or equal to\n0",
"currentValues": [
"1"
],
"actualValues": null,
The 'net.ipv4.conf.all.send_redirects' network parameter (/etc/sysctl.conf) allows ICMP routing redirection. If the system is not going to be used as a firewall or gateway to pass network traffic, and this parameter is not disabled, malicious users may attempt to spoof source addresses or redirect traffic to a host with a network sniffer, so this value should be set according to the needs of the business.

Remediation: "Set the following parameters in the /etc/sysctl.conf file: net.ipv4.conf.all.send_redirects = 0 OR Run the following commands to set the active kernel parameters: sysct1 -w net.ipv4.conf.all.send_redirects=0 sysct1 -w net.ipv4.route.flush=1."

Rationale: "The 'net.ipv4.conf.all.send_redirects' network parameter /etc/sysctl.conf allows ICMP routing redirection. If the system is not going to be used as a firewall or gateway to pass network traffic, and this parameter is not disabled, malicious users may attempt to spoof source addresses or redirect traffic to a host with a network sniffer, so this value should be set according to the needs of the business."
"customerId": "xx",
"assetId": xx,
"technology": {
  "id": xx,
  "name": "CentOS 7.x"
},
"criticality": {
  "label": "CRITICAL",
  "value": 4
},
"evidence": {
  "expectedValues": "Setting not found
------------ OR
------------
File not found
------------ OR ------------
matches regular expression list\n.*",
  "currentValues": [
    "Setting not found"
  ],
  "actualValues": null,
  "directoryFimUdc": null
},
"causeOfFailure": null,
"currentBatch": 1,
"totalBatches": 1
}

Get Posture info without lastEvaluationDate, without evidence, without compression, without lastScanDate

User Input: evidenceRequired=0 & compressionRequired=0

Request:

curl -X POST "https://gateway.<assigned URL>/pcrs/1.0/posture/postureInfo?evidenceRequired=0&compressionRequired=0" -H "accept: */*
-H "Authorization: Bearer <token>" -H "Content-Type: application/json" -d 
"[{"policyId":"xx","subscriptionId":"xx","hostIds":[]}]" 

Response:

[
  {
    "id": xx,
    "instance": "os",
    "policyId": "<POLICY ID>",
    "policyTitle": "<POLICY TITLE>",
    "netBios": "<NETBIOS>",
    "controlId": "<CONTROL ID>",
    "controlTitle": "<CONTROL TITLE>",
    "netBIOS": "<NETBIOS>",
    "netBIOSHostId": "<NETBIOSHOSTID>",
    "netBIOSGroupName": "<NETBIOSGROUPNAME>",
    "netBIOSHostGroup": "<NETBIOSHOSTGROUP>",
    "netBIOSCapabilities": "<NETBIOSCAPABILITIES>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>",
    "netBIOSOptions": "<NETBIOSOPTIONS>"}
"technologyId": "<TECHNOLOGY ID>",
"status": "Passed",
"previousStatus": "Passed",
"firstFailDate": "",
"lastFailDate": "",
"firstPassDate": "2021-10-14T11:19:31Z",
"lastPassDate": "2021-10-18T06:17:29Z",
"postureModifiedDate": "2021-10-14T11:19:30Z",
"lastEvaluatedDate": "2021-10-18T06:17:29Z",
"created": "2021-10-29T08:38:14Z",
"hostId": "<HOST ID>",
"ip": "xx.xx.xx.xx",
"trackingMethod": "IP",
"os": null,
"osCpe": "cpe:/o:cisco:asa:9.2%284%29:::",
"domainName": "<DOMAIN NAME>",
"dns": null,
"qgHostId": null,
"networkId": "0",
"networkName": "Global Default Network",
"complianceLastScanDate": "2021-10-14T09:37:38Z",
"customerUuid": "<CUSTOMER UUID>",
"customerId": "<CUSTOMER ID>",
"assetId": "<ASSET ID>",
"technology": {
   "id": xx,
   "name": "Cisco ASA 9.x"
},
"criticality": {
   "label": "MEDIUM",
   "value": 2
},
"evidence": null,
"causeOfFailure": null
},
{
   "id": xx,
   "instance": "os",
   "policyId": "<POLICY ID>",
   "policyTitle": "<POLICY TITLE>",
   "netBios": "<NETBIOS>",
   "controlId": "<CONTROL ID>",
   "technologyId": "<TECHNOLOGY ID>",
   "status": "Passed",
   "previousStatus": "Passed"
Compliance

Compliance Posture Information

"firstFailDate": "",
"lastFailDate": "",
"firstPassDate": "2021-10-25T07:21:13Z",
"lastPassDate": "2021-10-29T08:38:10Z",
"postureModifiedDate": "2021-10-25T07:21:11Z",
"lastEvaluatedDate": "2021-10-29T08:38:10Z",
"created": "2021-10-29T08:38:14Z",
"hostId": "<HOST ID>",
"ip": "xx.xx.xx.xx",
"trackingMethod": "IP",
"os": null,
"osCpe":
"cpe:/o:microsoft:windows_server_2012:r2::x64:",
"domainName": "<DOMAIN NAME>",
"dns": "comdevsql2016",
"qgHostid": null,
"networkId": "0",
"networkName": "Global Default Network",
"complianceLastScanDate": "2021-10-28T16:53:14Z",
"customerUuid": "0a387e70-8b26-78ff-8145-017b816fa17f",
"customerId": "<CUSTOMER ID>",
"assetId": "<ASSET ID>",
"technology": {
  "id": xx,
  "name": "Windows Server 2012 R2"
},
"criticality": {
  "label": "CRITICAL",
  "value": 4
},
"evidence": null,
"causeOfFailure": null
},
{
  "id": 19235413,
  "instance": "MSSQL 2016:1:1433:MSSQLSERVER:DB",
  "policyId": "<POLICY ID>",
  "policyTitle": "<POLICY TITLE>",
  "netBios": "<NETBIOS>",
  "controlId": "<CONTROL ID>",
  "technologyId": "<TECHNOLOGY ID>",
  "status": "Passed",
  "previousStatus": "Passed",
  "firstFailDate": "",
  "lastFailDate": "",
  "firstPassDate": "2021-10-28T16:53:06Z"},
Compliance Posture Information

"lastPassDate": "2021-10-29T08:38:10Z",
"postureModifiedDate": "2021-10-28T16:53:06Z",
"lastEvaluatedDate": "2021-10-29T08:38:10Z",
"created": "2021-10-29T08:38:13Z",
"hostId": "<HOST ID>",
"ip": "xx.xx.xx.xx",
"trackingMethod": "IP",
"os": null,
"osCpe":
"cpe:/o:microsoft:windows_server_2012:r2::x64:",
"domainName": "<DOMAIN NAME>",
"dns": "comdevsql2016",
"qgHostid": null,
"networkId": "0",
"networkName": "Global Default Network",
"complianceLastScanDate": "2021-10-28T16:53:14Z",
"customerUuid": "<CUSTOMER UUID>",
"customerId": "<CUSTOMER ID>",
"assetId": "<ASSET ID>",
"technology": {
   "id": xx,
   "name": "Microsoft SQL Server 2016"
},
"criticality": {
   "label": "MEDIUM",
   "value": 2
},
"evidence": null,
"causeOfFailure": null
}

Get Posture Info without lastEvaluationDate, with evidence, without compression, without lastScanDate

User input: evidenceRequired=1 & compressionRequired=0

Request:


Response:
[{
  "id": xx,
  "instance": "os",
  "policyId": "<POLICY ID>",
  "policyTitle": "<POLICY TITLE>",
  "netBios": "<NETBIOS>",
  "controlId": "<CONTROL ID>",
  "technologyId": "<TECHNOLOGY ID>",
  "status": "Passed",
  "previousStatus": "Passed",
  "firstFailDate": "",
  "lastFailDate": "",
  "firstPassDate": "2021-10-14T11:19:31Z",
  "lastPassDate": "2021-10-18T06:17:29Z",
  "postureModifiedDate": "2021-10-14T11:19:30Z",
  "lastEvaluatedDate": "2021-10-18T06:17:29Z",
  "created": "2021-10-29T08:40:38Z",
  "hostId": "<HOST ID>",
  "ip": "xx.xx.xx.xx",
  "trackingMethod": "IP",
  "os": null,
  "osCpe": "cpe:/o:cisco:asa:9.2%284%29:::",
  "domainName": "<DOMAIN NAME>",
  "dns": null,
  "qgHostid": null,
  "networkId": "0",
  "networkName": "Global Default Network",
  "complianceLastScanDate": "2021-10-14T09:37:38Z",
  "customerUuid": "<CUSTOMER UUID>",
  "customerId": "<CUSTOMER ID>",
  "assetId": "<ASSET ID>",
  "technology": {
    "id": xx,
    "name": "Cisco ASA 9.x"
  },
  "criticality": {
    "label": "MEDIUM",
    "value": 2
  },
  "evidence": {
    "expectedValues": "\nFilter 2 not found:
^\[\*\*.\d\]n-------- OR --------\nFilter 1 not found:
show clock detail\n-------- OR --------\nmatches
regular expression list\n.n.*",
    "currentValues": [

"show clock detail:08:26:29.074 pdt Thu Oct 14 2021",

},
  "actualValues": null,
  "directoryFimUdc": null
},
  "causeOfFailure": null
}],
{
  "id": "xx",
  "instance": "MSSQL 2016:1:1433:MSSQLSERVER:DB",
  "policyId": "<POLICY ID>",
  "policyTitle": "<POLICY TITLE>",
  "netBios": "<NETBIOS>",
  "controlId": "<CONTROL ID>",
  "technologyId": "xx",
  "status": "Passed",
  "previousStatus": "Passed",
  "firstFailDate": "",
  "lastFailDate": "",
  "firstPassDate": "2021-10-28T16:53:06Z",
  "lastPassDate": "2021-10-29T08:39:07Z",
  "postureModifiedDate": "2021-10-28T16:53:06Z",
  "lastEvaluatedDate": "2021-10-29T08:39:07Z",
  "created": "2021-10-29T08:40:46Z",
  "hostId": "<HOST ID>",
  "ip": "xx.xx.xx.xx",
  "trackingMethod": "IP",
  "os": null,
  "osCpe":
  "cpe:/o:microsoft:windows_server_2012:r2::x64:",
  "domainName": "<DOMAIN NAME>",
  "dns": "comdevsql2016",
  "qgHostid": null,
  "networkId": "0",
  "networkName": "Global Default Network",
  "complianceLastScanDate": "2021-10-28T16:57:58Z",
  "customerUuid": "<CUSTOMER UUID>",
  "customerId": "<CUSTOMER ID>",
  "assetId": "<ASSET ID>",
  "technology": {
    "id": "xx",
    "name": "Microsoft SQL Server 2016"
  },
  "criticality": {
    "label": "MEDIUM",
"value": 2
},
"evidence": {
  "expectedValues": "\nSet status to PASS if no data
found\n---------- OR ----------\nmatches regular expression
list\n.*",
  "currentValues": [
    "Error Code 35:Failed to execute database query"
  ],
  "actualValues": null,
  "directoryFimUdc": null
},
"causeOfFailure": null
]

Get Posture Info without lastEvaluationDate, without evidence, with compression, with lastScanDate

User input: evidenceRequired=0 & compressionRequired=1 & lastScanDate=yyyy-mm-dd:Thh:mm:ssZ

Request:
Curl-X POST
"https://qualysapi.qualys.com/pcrs/1.0/posture/postureInfo?evidenceRequired=0&compressionRequired=1&lastEvaluationDate=2021-12-17T18:48:16Z&lastScanDate=2021-12-17T18:48:16Z" -H
"accept: */*" -H
"Content-Type: application/json" -d "
[
{"policyId": "<POLICY ID>", "subscriptionId": "<SUBSCRIPTION ID>", "hostIds": ["<HOST ID>"]}
]

Response:
JSON output:
[
  {
    "id": <HOST INSTANCE ID>,
    "instance": "os",
    "policyId": "<POLICY ID>",
    "policyTitle": "<POLICY TITLE>",
    "netBios": "<NETBIOS>",
    "controlId": <CONTROL ID>,
    "controlStatement": "Status of the 'Minimum Password Length' setting",
    "rationale": "Among the several characteristics that make 'user identification' via password a secure and workable solution is setting a 'minimum password length' requirement. Each character that is added to the password length squares the difficulty of breaking the password via"
'brute force,' which attempts using every combination possible within the password symbol set-space, in order to discover a user's password. While no 'minimum length' can be guaranteed secure, eight (8) is commonly considered to be the minimum for most application access, along with requiring other password security factors, such as increasing the size of the symbol set-space by requiring mixed-cases, along with other forms of password variability creation, increases the difficulty of breaking any password by brute-force attack.

"remediation": "To specify password length requirements for new accounts, edit the file \"/etc/login.defs\" and add or correct the following lines: 

PASS_MIN_LEN: <required value>

example: \nPASS_MIN_LEN 14 

Note: The DoD requirement is \"14\". If a program consults \"/etc/login.defs\" and also another PAM module (such as \"pam_cracklib\") during a password change operation, then the most restrictive must be satisfied.\",

"controlReference": null,
"technologyId": <TECHNOLOGY ID>,
"status": "Passed",
"previousStatus": "Passed",
"firstFailDate": ",
"lastFailDate": ",
"firstPassDate": "2021-12-23T08:20:23Z",
"lastPassDate": "2022-02-02T11:54:20Z",
"postureModifiedDate": "2021-12-23T08:22:22Z",
"lastEvaluatedDate": "2022-02-02T11:54:20Z",
"created": "2022-07-11T11:53:46Z",
"hostId": <HOST ID>,
"CLOUD_RESOURCE_ID": "<CLOUD RESOURCE ID>",
"ip": "xx.xx.xx.xxx",
"trackingMethod": "EC2",
"os": "Red Hat Enterprise Linux 8.3",
"osCpe": null,
"domainName": "<DOMAIN NAME>",
"dns": "ip-xx-xx-xx-xxx.af-south-1compute.internal",
"qgHostid": null,
"networkId": 0,
"networkName": "Global Default Network",
"complianceLastScanDate": "2021-12-23T12:59:04Z",
"customerUuid": "<CUSTOMER UUID>",
"customerId": "<CUSTOMER ID>",
"assetId": <ASSET ID>,
"technology": {
  "id": 217,
  "name": "Red Hat Enterprise Linux 8.x"
},
"criticality": {
  "label": "CRITICAL",
  "value": 4
},
"evidence": null,
"causeOfFailure": null,
"currentBatch": 8,
"totalBatches": 12}
Get Posture Info without evidence, without compression, with statusChangedSince

Request:

curl -X POST
"https://gateway.<assigned URL>/pcrs/1.0/posture/postureInfo?evidenceRequired=0&compressionRequired=0&statusChangedSince=2021-12-23" -H "accept: */*" -H "Authorization: Bearer <token>" -H "Content-Type: application/json" -d "[{"policyId":"<POLICYID>","subscriptionId":"<SUBSCRIPTIONID>","hostIds":["<HOST ID1>","<HOST ID2>"]}]"

Response:

JSON output:

[

{
"id": 24705485,
"instance": "os",
"policyId": <POLICY ID>,
"policyTitle": "pcas_win16_redhat7 tech",
"netBios": "<NETBIOS>",
"controlId": 1071,
"controlStatement": "Status of the 'Minimum Password Length' setting",
"rationale": "Among the several characteristics that make 'user identification' via password a secure and workable solution is setting a 'minimum password length' requirement. Each character that is added to the password length squares the difficulty of breaking the password via 'brute force,' which attempts using every combination possible within the password symbol set-space, in order to discover a user's password. While no 'minimum length' can be guaranteed secure, eight (8) is commonly considered to be the minimum for most application access, along with requiring other password security factors, such as increasing the size of the symbol set-space by requiring mixed-cases, along with other forms of password variability creation, increases the difficulty of breaking any password by brute-force attack.",
"remediation": "To establish the recommended configuration via GP, set the following UI path to 14 or more character(s):\n\n\nComputer Configuration\n\nConfiguration\n\nWindows Settings\n\nSecurity Settings\n\nAccount Policies\n\nPassword Policy\n\nMinimum password length",
"controlReference": null,
"technologyId": 106,
"status": "Passed",
"previousStatus": "Passed",
"firstFailDate": "",
"lastFailDate": ""}
Compliance
Compliance Posture Information

"firstPassDate": "2022-11-09T12:50:12Z",
"lastPassDate": "2022-12-06T06:42:21Z",
"postureModifiedDate": "2022-11-09T12:50:12Z",
"lastEvaluatedDate": "2022-12-06T06:42:21Z",
"created": "2022-12-07T07:35:56Z",
"hostId": <HOST ID>,
"CLOUD_RESOURCE_ID": null,
"ip": "xx.xx.xx.xxx",
"trackingMethod": "IP",
"os": null,
"osCpe": "cpe:/o:microsoft:windows_server_2016:1607::x64:",
"domainName": "<DOMAIN NAME>",
"dns": "<DNS>",
"qgHostid": null,
"networkId": 0,
"networkName": "Global Default Network",
"complianceLastScanDate": "2022-08-23T04:57:05Z",
"customerUuid": "<CUSTOMER UUID>",
"customerId": "<CUSTOMER ID>",
"assetId": <ASSET ID>,
"technology": {
  "id": 106,
  "name": "Windows 2016 Server"
},
"criticality": {
  "label": "high updated",
  "value": 5
},
"evidence": {
  "expectedValues": "\nAttribute not found\n------------
OR ------------
greater than or equal to\n0",
  "currentValues": [
    "6"
  ],
  "actualValues": null,
  "directoryFimUdc": null
},
"causeOfFailure": null,
"currentDataSizeKB": "2.41",
"totalDataSizeKB": "2.41",
"currentBatch": 1,
"totalBatches": 1
}
Control Criticality

Control Criticality is a feature in Policy Compliance that provides ratings for controls, including the ability to customize ratings at the control level and at the policy level. Several APIs include control criticality in the API output.

Control Criticality must be enabled in your account — By default, control criticality will not be enabled while we are updating the default criticality settings in the control library. If you want this feature, please contact Support or your Technical Account Manager.
Exceptions

/api/2.0/fo/compliance/exception/

[GET]  [POST]

List, request, update and delete exceptions in your account. Supported method differs per request type, i.e. list, create etc).

The Exception API is only available if you have Policy Compliance (PC) module enabled for your subscription. Non Manager users must be granted this permission in their account settings.

User Permissions

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>List, request, update, delete exceptions for all hosts in subscription.</td>
</tr>
<tr>
<td>Auditor</td>
<td>List, request, update, delete exceptions for all hosts in subscription.</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>List, request, update, delete exceptions for hosts in their assigned business unit.</td>
</tr>
<tr>
<td>Scanner, Reader</td>
<td>List, request, update exceptions for hosts in their account. Updates are limited to adding comments.</td>
</tr>
</tbody>
</table>

List exceptions

By default, all exceptions in the user’s account are listed. Use the optional parameters to filter the list output.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>exception_number={value}</td>
<td>(Optional) Show a specific exception by specifying a valid exception number.</td>
</tr>
<tr>
<td>ip={value}</td>
<td>(Optional) Show exceptions associated with a specific host by specifying a host IP address. You may enter individual IP address that belong to the Policy Compliance module.</td>
</tr>
<tr>
<td>network_name={value}</td>
<td>(Optional) Show exceptions for a particular network by specifying the network name.</td>
</tr>
<tr>
<td>status={value}</td>
<td>(Optional) Show exceptions with specified status value: pending, approved, rejected or expired. Tell me about exception status</td>
</tr>
<tr>
<td>control_id={value}</td>
<td>(Optional) Show exceptions for a specific control by specifying valid control ID. If the value is set to 23, the matching control IDs may include 23, 234, 2343, 233.</td>
</tr>
<tr>
<td>control_statement={value}</td>
<td>(Optional) Show exceptions for certain controls associated with a certain policy by specifying control statement. Partial control statement is also valid.</td>
</tr>
</tbody>
</table>
Tell me about exception status

Pending - An exception is in a Pending state when first requested by a user. Also, if a previously accepted or rejected exception is reopened, then it goes back to Pending.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy_id={value}</td>
<td>(Optional) Show exceptions for controls associated with a certain policy by specifying a valid policy ID.</td>
</tr>
<tr>
<td>technology_name={value}</td>
<td>(Optional) Show exceptions for controls with a certain technology by specifying the technology name.</td>
</tr>
<tr>
<td>assignee_id={value}</td>
<td>(Optional) Show exceptions with a certain assignee by specifying an assignee’s user ID.</td>
</tr>
<tr>
<td>created_by={value}</td>
<td>(Optional) Show exceptions that were created by a particular user by specifying the user ID.</td>
</tr>
<tr>
<td>modified_by={value}</td>
<td>(Optional) Show exceptions that were modified by a particular user by specifying the user ID.</td>
</tr>
</tbody>
</table>
| details={Basic|All|None} | (Optional) Show the requested amount of information for each control. A valid value is:  

None - Only exception numbers.  
Basic (default) - All details except comments history.  
All - All details including comments history. |
| is_active={0|1}             | (Optional). Show only exceptions that are active or inactive in the output. Specify 1 to show only active exceptions. Specify 0 to show only inactive exceptions. When unspecified, both active and inactive exceptions are shown. |
| created_after_date={mm/dd/yyyy} | (Optional) Show exceptions created (requested) after the specified date. The valid date format is mm/dd/yyyy. |
| updated_after_date={mm/dd/yyyy} | (Optional) Show exceptions that were updated after the specified date. The valid date format is mm/dd/yyyy. |
| expired_before_date={mm/dd/yyyy} | (Optional) Show exceptions that will expire before the specified date. The valid date format is mm/dd/yyyy. |
| expired_after_date={mm/dd/yyyy} | (Optional) Show exceptions that will expire after the specified date. The valid date format is mm/dd/yyyy. |
| exception_numbers={value}  | (Optional) Show a specific exception by specifying a valid exception number. Multiple entries are comma separated. An exception number range is specified with a hyphen (for example, 289-292). |
| exception_number_min={value} | (Optional) Show only exceptions that have a exception number greater than or equal to the specified value. |
| exception_number_max={value} | (Optional) Show only exceptions that have exception number less than or equal to the specified value. |
| truncation_limit={value}   | (Optional) Specify the maximum number of exceptions to be listed per request. When not specified, the truncation limit is set to 1000 records. You may specify a value less than the default (1-999) or greater than the default (1001-1000000). |
Approved - An exception is in an Approved state when it is reviewed and accepted by an authorized user. You would accept an exception if it’s determined that the host should be exempt from the specified control. As long as the host is exempt for the control, a status of PassedE appears in compliance reports. The status changes back to Failed when the exception expires.

Rejected - An exception is in a Rejected state when it is reviewed and rejected by an authorized user. You would reject an exception if it’s determined that the host should not be exempt from the specified control. When an exception is rejected, a status of Failed continues to appear for the host/control in compliance reports.

Expired - An exception is in an Expired state when the exception was previously accepted but the time limit has been reached. When an exception is expired, a status of Failed appears again for the host/control in compliance reports.

Sample - List exceptions with failed status

API request:
```
curl -s -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl demo 2"
-D headers.15
"https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/?
action=list&policy_id=1174&status=Failed"
```

XML response:
```
<?xml version="1.0" encoding="UTF-8" ?>
"https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/posture_info_list_output.dtd">...
  <INFO>
    <ID>1174</ID>
    <HOST_ID>563352</HOST_ID>
    <CONTROL_ID>1072</CONTROL_ID>
    <TECHNOLOGY_ID>2</TECHNOLOGY_ID>
    <INSTANCE></INSTANCE>
    <STATUS>Failed</STATUS>
    <POSTURE_MODIFIED_DATE>2015-09-02T08:16:33Z</POSTURE_MODIFIED_DATE>
  </INFO>
...
```

Sample - List exception number, show all details

API request:
```
curl -s -u "USERNAME:PASSWORD" -H "X-Requested-With: curl demo 2"
-D headers.15
"https://qualysapi.qualys.com/api/2.0/fo/compliance/exception/?action=list&exception_number=58&details=All"
```

816
Compliance
Exceptions

XML response:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
"https://qualysapi.qualys.com/api/2.0/fo/compliance/exception/
exception_list_output.dtd">
<EXCEPTION_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-01-15T11:26:34Z</DATETIME>
    <EXCEPTION_LIST>
      <EXCEPTION>
        <EXCEPTION_NUMBER>58</EXCEPTION_NUMBER>
        <HOST>
          <IP_ADDRESS>10.10.30.159</IP_ADDRESS>
        </HOST>
        <TECHNOLOGY>
          <ID>11</ID>
          <NAME><![CDATA[Red Hat Enterprise Linux 5.x]]></NAME>
        </TECHNOLOGY>
        <POLICY>
          <ID>789422824</ID>
          <NAME><![CDATA[RHEL 5.x]]></NAME>
        </POLICY>
        <CONTROL>
          <CID>1073</CID>
          <STATEMENT><![CDATA[Status of the 'Maximum Password Age' setting
          (expiration) / Accounts having the 'password never expires'
          flag set]]></STATEMENT>
          <CRITICALITY>
            <VALUE>5</VALUE>
            <LABEL><![CDATA[URGENT]]></LABEL>
          </CRITICALITY>
        </CONTROL>
        <ASSIGNEE><![CDATA[Scanner User]]></ASSIGNEE>
        <STATUS>Rejected</STATUS>
        <ACTIVE>1</ACTIVE>
        <REOPEN_ON_EVIDENCE_CHANGE>0</REOPEN_ON_EVIDENCE_CHANGE>
        <EXPIRATION_DATE>N/A</EXPIRATION_DATE>
        <MODIFIED_DATE>2017-01-15T08:53:19Z</MODIFIED_DATE>
        <HISTORY_LIST>
          <HISTORY>
            <USER><![CDATA[John (mnc_su)]]></USER>
            <COMMENT><![CDATA[test]]></COMMENT>
            <INSERTION_DATE>2017-01-05T06:48:13Z</INSERTION_DATE>
          </HISTORY>
        </HISTORY_LIST>
      </EXCEPTION>
    </EXCEPTION_LIST>
  </RESPONSE>
</EXCEPTION_LIST_OUTPUT>
```
An exception is created with the expiry date matching the creation date. You can update the exception to change it.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=request</td>
<td>(Required) POST method must be used. action=create is also valid.</td>
</tr>
<tr>
<td>control_id={value}</td>
<td>(Required) Specify the control ID of the control for which you want to request an exception.</td>
</tr>
<tr>
<td>host_id={value}</td>
<td>(Required) Specify the host ID of the host for which you want to request an exception.</td>
</tr>
<tr>
<td>policy_id={value}</td>
<td>(Required) Specify the policy ID of the policy that contains the control for which you want to request an exception.</td>
</tr>
<tr>
<td>technology_id={value}</td>
<td>(Required) Specify the technology ID of the technology associated with the host for which you want to request an exception.</td>
</tr>
<tr>
<td>instance_string={value}</td>
<td>(Optional) Specifies a single instance on the selected host. The instance string may be &quot;os&quot; or a string like &quot;oracle10:1:1521:ora10204u&quot;. This parameter must be specified with: host_id.</td>
</tr>
<tr>
<td>assignee_id={value}</td>
<td>(Required) You can assign exception to another user. Specify user ID of the user, who has access to the hosts that the exceptions apply to.</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Required) User defined comments.</td>
</tr>
<tr>
<td>reopen_on_evidence_change=0</td>
<td>(Optional) This applies only if the exception is approved. Reopen the exception if a future scan returns a value that is different than the current value and the control is still failing.</td>
</tr>
<tr>
<td>reopen_on_evidence_change=1</td>
<td></td>
</tr>
</tbody>
</table>
Sample - Request exception

API request:
```
curl -u "USERNAME:PASSWD" -H "X-Requested-WithError: Curl" -X "POST" -d "action=request&control_id=1113&host_id=28595192824&policy_id=801459496&technology_id=45&assignee_id=2449482824&reopen_on_evidence_change=1&comments=new exception" "https://qualysapi.qualys.com/api/2.0/fo/compliance/exception/
```

XML response:
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2015-12-15T10:14:43Z</DATETIME>
    <TEXT>Exception created successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>EXCEPTION_NUMBER</KEY>
        <VALUE>15</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

DTD
```
<platform API server>/api/2.0/fo/compliance/exception/
```

Update exceptions
You can make changes to one or more exceptions on your hosts. All the actions you take are logged in the exception history with your name and a time stamp for when the action took place.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=update</td>
<td>(Required) POST method must be used.</td>
</tr>
<tr>
<td>exception_numbers={value}</td>
<td>(Required) Show a specific exception by specifying a valid exception number. Multiple entries are comma separated. An exception number range is specified with a hyphen (for example, 50-55).</td>
</tr>
<tr>
<td>comments={value}</td>
<td>(Required) User defined comments. Your comments are saved in the exception history.</td>
</tr>
</tbody>
</table>
Compliance Exceptions

### Sample - Update exception

**API request:**

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=update&exception_numbers=55&status=Approved&end_date=12/16/2015&comments=status change"
```

**XML response:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/exception/exception_batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-01-07T11:24:42Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <NUMBER_SET>
          <NUMBER>55</NUMBER>
        </NUMBER_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

**DTD**

```
<platform API server>/api/2.0/fo/compliance/exception/exception_batch_return.dtd
```
Delete exceptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=delete</td>
<td>(Required) POST method must be used.</td>
</tr>
<tr>
<td>exception_numbers={value}</td>
<td>(Required) Specify the exception number. Enter one or more exception numbers and/or ranges. Multiple entries are comma separated.</td>
</tr>
</tbody>
</table>

Sample - Delete exceptions

API request:
```bash
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d "action=delete&exception_numbers=40-41" "https://qualyapi.qualys.com/api/2.0/fo/compliance/exception/
```

XML response:
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM "https://qualysapi.qualys.com/api/2.0/fo/compliance/exception/exception_batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2018-01-07T11:22:20Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Exception(s) deleted successfully</TEXT>
        <NUMBER_SET>
          <NUMBER_RANGE>40-41</NUMBER_RANGE>
        </NUMBER_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

DTD
`<platform API server>/api/2.0/fo/compliance/exception/exception_batch_return.dtd`
SCAP Cyberscope Report

Under the Federal Information Security Management Act of 2002 (FISMA), government agencies are obliged to report on their information security statuses using a common tool called Cyberscope. Qualys customers with the SCAP module enabled can scan their network and generate Cyberscope compatible XML reports, using new API functions, to meet these requirements.

Qualys provides 3 different API functions for generating Cyberscope compatible XML reports as described below. The Cyberscope reports generated using these API functions return XML output in LASR format.

Cyberscope report specification and the LASR format:
http://scap.nist.gov/use-case/cyberscope

SCAP Scan Results
/api/2.0/fo/asset/host/cyberscope/fdcc/scan/

Create a Cyberscope report using scan results for a particular SCAP scan in the user’s account. An SCAP scan ID or scan reference is required as input. The service uses only the data in the raw scan results to generate the report. When the parameters organisation_name1, organisation_name2, and organisation_name3 are specified, the <ai:Organization> elements are included in the XML report.

Permissions: Users have permission to run this API function when the SCAP module is enabled for the user’s subscription. Sub-accounts (Unit Managers, Scanners and Readers) must have the "Manage compliance" permission.

Sample 1 - Select SCAP Scan by Scan ID

Use the scan_id parameter to select an SCAP scan by scan ID. (A scan ID or reference number is required.)

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdcc/scan/?scan_id=4244823&organisation_name1=Name1&organisation_name2=Name2&organisation_name3=Name3"

To obtain the SCAP scan ID, log into the Qualys application and go to PC/SCAP > Scans > SCAP Scans to view the SCAP scans in your account. Hover over the SCAP scan that you’re interested in and view the scan results (select View from the Quick Actions menu). You’ll see the scan results URL in your browser and the scan ID value appears in the "id" parameter, as shown in this sample URL:

Sample 2 - Select SCAP Scan by Scan Reference

Use the scan_ref parameter to select an SCAP scan by scan reference number. (A scan reference number or scan ID is required.)

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdc
c/scan/?scan_ref=qscap/1337984725.4360&organisation_name1=Name1&organisation_name2=Name2&organisation_name3=Name3"

Sample 3 - IPs Filter

Use the optional ips parameter to include only certain IP addresses in the report. You can enter a single IP, multiple IPs and/or IP ranges. Multiple entries are comma separated.

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdc
c/scan/?scan_id=4268027&ips=10.10.26.183&organisation_name1=Name1&organisation_name2=Name2&organisation_name3=Name3"

SCAP Policy Results

/api/2.0/fo/asset/host/cyberscope/fdcc/policy/

Create a Cyberscope report using scan results data saved for a particular SCAP policy in the user’s account. A policy ID is required as input. These parameters allow users to customize the required “OrganisationName” elements in the XML report: organisation_name1, organisation_name2, and organisation_name3.

The service uses automatic SCAP policy data for a selected policy and reports this in the datapoint `<sr:DataPoint id="configuration_management_agency_deviations">`. The services uses the evidence data for the special rule "security_patches_up_to_date" and reports this in the datapoint `<sr:DataPoint id="vulnerability_management_product_vulnerabilities">`.

Permissions: Users have permission to run this API function when the SCAP module is enabled for the user’s subscription and sub-accounts (Unit Managers, Scanners and Readers) have the "Manage compliance" permission.

Sample 1 - Select an SCAP Policy

Use the policy_id parameter to select an SCAP policy. Hosts in the policy will be included in the report unless filters are specified using the parameter ips and/or as_ids.

curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdcc/policy/?policy_id=30231&organisation_name1=Name1&organisation_name2=Name2&organisation_name3=Name3"
To obtain the SCAP policy ID, log into the Qualys application and go to PC/SCAP > Policies to view the policies in your account. Hover over the SCAP policy that you’re interested in and edit it (select Edit from the Quick Actions menu). You’ll see the policy editor URL in your browser and the policy ID value appears in the “id” parameter, as shown in this sample URL:

https://qualyguard.qualys.com/fo/fdcc/edit_policy.php?id=12345&refresh_parent=1

Sample 2 - IPs Filter

Use the ips parameter to include only hosts with the specified IP addresses. Enter a single IP, multiple IPs and/or IP ranges using the ips parameter. Multiple entries are comma separated.

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdcc/policy/?policy_id=17012&ips=10.10.24.10&organisation_name1=Name1&organisation_name2=Name2&organisation_name3=Name3"
```

Sample 3 - Asset Groups Filter

Use the `as_ids` parameter to include only hosts in the specified asset groups. Multiple asset group IDs are comma separated.

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" "https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdcc/policy/?policy_id=17012&ag_ids=397405&ips=10.10.25.70&organisation_name1=Name1&organisation_name2=Name2&organisation_name3=Name3"
```

**SCAP Global Results**

`/api/2.0/fo/asset/host/cyberscope/`

Create a Cyberscope report using the SCAP scan data saved for all the SCAP policies in the subscription and also the automatic VM scan data saved in the subscription. You must enter IPs/ranges and/or asset group IDs as input. These parameters allow users to customize the required "OrganisationName" elements in the XML report: organisation_name1, organisation_name2, and organisation_name3.

The service uses SCAP scan data for all the SCAP policies in the subscription and reports this in the datapoint `<sr:DataPoint id:”configuration_management_agency_deviations”>`. This datapoint will include multiple Benchmark Data sections, one for each policy. Also the service uses the automatic VM data for applicable IPs (IPs in SCAP policies) and reports this in the datapoint `<sr:DataPoint id:"vulnerability_management_product_vulnerabilities”>`.

Permissions: Users have permission to run this API function when the SCAP module is enabled for the user’s subscription. Sub-accounts (Unit Managers, Scanners, and Readers) will view only data for IP addresses that their accounts have access to.
Sample 1 - Select Hosts by IP
Use the ips parameter to select hosts by IP/range. You can enter a single IP, multiple IPs and/or IP ranges using the ips parameter. Multiple entries are comma separated. (This parameter and/or ag_ids is required.)

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/?ips=10.10.24.52&organisation_name1=Name1&organisation_name2=Name2&organisation_name3=Name3"
```

Sample 2 - Select Hosts by Asset Group
Use the as_ids parameter to select hosts by asset group ID. You can enter one or more asset group IDs. Multiple IDs are comma separated. (This parameter and/or ips is required.)

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/?ag_ids=503424&organisation_name1=Name1&organisation_name2=Name2&organisation_name3=Name3"
```

It's possible to select hosts by entering a combination of IPs/ranges and asset group IDs.

```bash
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/?ips=10.10.24.52,10.10.25.2-10.10.25.255&ag_ids=503424,503430&organisation_name1=Name1&organisation_name2=Name2&organisation_name3=Name3"
```
SCAP ARF Report
/api/2.0/fo/compliance/scap/arf/

Create a SCAP scan report in Asset Reporting Format (ARF), a requirement in the SCAP 1.2 Specifications from NIST.

Permissions - Users have permission to run this API function when the SCAP module is enabled for the user’s subscription. Sub-accounts (Unit Managers, Scanners and Readers) must have the "Manage compliance" permission.

Input parameters:

Parameter | Description
--- | ---
scan_id={value} (Required) | The scan ID for a finished SCAP scan.
ips={value} (Optional) | Use this parameter if you want to include only certain IP addresses in the report. You can enter a single IP, multiple IPs and/or ranges. Multiple entries are comma separated.
ips_network_id={value} (Optional and valid only when the Network Support feature is enabled and the policy has SCAP 1.2 content) | Use this parameter to restrict the report’s target to the IPs specified in the “ips” parameter (“ips_network_id” is valid only when “ips” is specified in the same request).

How do I find the scan ID? You’ll see the scan ID in the Qualys user interface, when viewing SCAP scan results. In the scan results window’s title bar you’ll see the report URL with its ID number in the “id” parameter, like this:
https://qualyguard.qualys.com/fo/report/fdcc/fdcc_scan_result.php?id=3362251

API Request:
```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X POST -d "scan_id=3362251&ips=10.10.10.1-10.10.10.10" "https://qualysapi.qualys.com/api/2.0/fo/compliance/scap/arf/"
```

XML Output:
The XML output is compliant with the ARF 1.1 Schema. Show me this schema
SCAP Policy List

/api/2.0/fo/compliance/fdcc_policy/?action=list

[GET] [POST]

View a list of SCAP policies visible to the user. Optional input parameters support filtering the policy list output.

Maximum Policies per API Request

A maximum of 1,000 SCAP policy records can be processed per request. If the requested list identifies more than 1,000 policies, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of policy records.

Permissions

<table>
<thead>
<tr>
<th>User Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>View all SCAP policies in subscription. View asset group information for all asset groups assigned to policies.</td>
</tr>
<tr>
<td>Auditor</td>
<td>View all SCAP policies in subscription. View asset group information for all asset groups assigned to policies.</td>
</tr>
<tr>
<td>Unit Manager</td>
<td>View all SCAP policies in subscription, when the &quot;Manage compliance&quot; permission is turned on in the user account settings. View asset group information for asset groups assigned to SCAP policies, when the user has permission to view these asset groups.</td>
</tr>
<tr>
<td>Scanner</td>
<td>View all SCAP policies in subscription, when the &quot;Manage compliance&quot; permission is turned on in the user account settings. View asset group information for asset groups assigned to SCAP policies, when the user has permission to view these asset groups.</td>
</tr>
<tr>
<td>Reader</td>
<td>View all SCAP policies in subscription, when the &quot;Manage compliance&quot; permission is turned on in the user account settings. View asset group information for asset groups assigned to SCAP policies, when the user has permission to view these asset groups.</td>
</tr>
</tbody>
</table>

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>echo_request=[0</td>
<td>1]</td>
</tr>
</tbody>
</table>
Compliance
SCAP Policy List

Sample - SCAP Policy List
Sample SCAP policy list output (fragment) with details=All is below.

```xml
<!DOCTYPE POLICY_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/fdcc_policy/fdcc_policy_list_output.dtd">

<FDCC_POLICY_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2012-07-19T22:10:16Z</DATETIME>
    <FDCC_POLICY_LIST>
      <FDCC_POLICY>
        <ID>10235</ID>
        <TITLE><![CDATA[XP policy]]></TITLE>
        <DESCRIPTION><![CDATA[This benchmark has been created to assist IT professionals, in particular Windows XP system administrators and information security personnel, in effectively securing Windows XP Professional SP2 systems.]]></DESCRIPTION>
        <BENCHMARK>![CDATA[FDCC-Windows-XP]]</BENCHMARK>
        <BENCHMARK_PROFILE>![CDATA[federal_desktop_core_configuration_version_1.2.1.0]]</BENCHMARK_PROFILE>
        <BENCHMARK_STATUS_DATE>2009-04-08T00:00:00Z</BENCHMARK_STATUS_DATE>
        <VERSION>![CDATA[v1.2.1.0]]</VERSION>
        <TECHNOLOGY>![CDATA[Windows XP Desktop]]</TECHNOLOGY>
        <NIST_PROVIDED>![CDATA[No]]</NIST_PROVIDED>
      </FDCC_POLICY>
    </FDCC_POLICY_LIST>
  </RESPONSE>
</FDCC_POLICY_LIST_OUTPUT>
```

Parameter | Description
---|---
details=**Basic**| (Optional) Show the requested amount of host information for each host. A valid value is: Basic - (default) Includes all SCAP policy details except the asset group list and SCAP file list. All - includes all SCAP policy details. None - includes SCAP policy ID and title.
details=**All**| (Optional) Show only certain SCAP policy IDs/ranges. One or more policy IDs/ranges may be specified. Valid host IDs are required. Multiple entries are comma separated. A policy ID range is specified with a hyphen (for example, 190-400).
details=**None**| (Optional) Show only SCAP policies which have a minimum SCAP policy ID value. A valid SCAP policy ID is required.
details=**None**| (Optional) Show only SCAP policies which have a maximum SCAP policy ID value. A valid SCAP policy ID is required.

DTD

```xml
<platform API server>/api/2.0/fo/compliance/fdcc_policy/fdcc_policy_list_output.dtd
```

Sample - SCAP Policy List
Sample SCAP policy list output (fragment) with details=All is below.
<ASSET_GROUP_LIST>
  <ASSET_GROUP>
    <ID>414242</ID>
    <TITLE><![CDATA[10.10.10.40]]></TITLE>
  </ASSET_GROUP>
  <ASSET_GROUP>
    <ID>414942</ID>
    <TITLE><![CDATA[10 range]]></TITLE>
  </ASSET_GROUP>
  <ASSET_GROUP>
    <ID>419582</ID>
    <TITLE><![CDATA[10.10.10.29]]></TITLE>
  </ASSET_GROUP>
  <ASSET_GROUP>
    <ID>419702</ID>
    <TITLE><![CDATA[10.10.10.28-16-191]]></TITLE>
  </ASSET_GROUP>
</ASSET_GROUP_LIST>

<FDCC_FILE_LIST>
  <FDCC_FILE>
    <FILE_NAME><![CDATA[fdcc-winxp-xccdf.xml]]></FILE_NAME>
    <FILE_HASH><![CDATA[0c1a49c4ca47187995b543cfdfc35783]]></FILE_HASH>
  </FDCC_FILE>
  <FDCC_FILE>
    <FILE_NAME><![CDATA[fdcc-winxp-cpe-oval.xml]]></FILE_NAME>
    <FILE_HASH><![CDATA[f397b9068b3881ef2a35c948326e6e4e]]></FILE_HASH>
  </FDCC_FILE>
  <FDCC_FILE>
    <FILE_NAME><![CDATA[fdcc-winxp-cpe-dictionary.xml]]></FILE_NAME>
    <FILE_HASH><![CDATA[33b9b03961c58e65263bc86b4e0cdef]]></FILE_HASH>
  </FDCC_FILE>
</FDCC_FILE_LIST>
<FDCC_FILE>
  <FILE_NAME><![CDATA[fdcc-winxp-oval.xml]]></FILE_NAME>
  <FILE_HASH><![CDATA[d1cf1f195bb58f295ca4b17dea2f99f0]]></FILE_HASH>
</FDCC_FILE>

<FDCC_FILE>
  <FILE_NAME><![CDATA[fdcc-winxp-patches.xml]]></FILE_NAME>
  <FILE_HASH><![CDATA[4ae1b306344ef564c5da479a4a3d7f53]]></FILE_HASH>
</FDCC_FILE>
</FDCC_FILE_LIST>
</FDCC_POLICY>

...<FDCC_POLICY_LIST>
...
<FDCC_POLICY_LIST_OUTPUT>
Users and Activity Log

Add, update, list and delete users in your subscription.

User List
Add/Edit User
User Registration Process
Accept Qualys EULA
Activate/Deactivate Users
User Password Change
Export User Activity Log

User List

/msp/user_list.php

[GET] [POST]

View the users in the subscription. XML responses provides details about each user such as the user’s login ID, account info, assigned asset groups, permissions. Session based authentication is not supported using this API.

When the API request is made by a Manager or Unit Manager, the last login date for each user is provided in the XML results. This is the most recent date and time the user logged into the service. For a Manager, the last login date appears for all users in the subscription. For a Unit Manager, the last login date appears for all users in the Unit Manager’s same business unit.

Permissions - Managers and Administrators can view all users in subscription. See Unit Manager Permissions for full details.

Express Lite - This API is available to Express Lite users.

Unit Manager Permissions

Unit Managers can view full user account details for users in their business unit. Unit Managers may also be able to view partial user account details for users outside of their business unit. This is determined by a subscription level permission set by Managers in the user interface.

If “Restrict view of user information for users outside of business unit” is not selected (the default), then Unit Managers have an unrestricted view and can see partial details about users who are not in their assigned business unit.
If “Restrict view of user information for users outside of business unit” is selected, then Unit Managers have a restricted view and cannot see any details for users who are not in their assigned business unit. For example, Unit Managers in Business Unit A would not be able to view general information or asset group assignments for users in Business Unit B.

The following table describes the amount of detail visible to Unit Managers for different types of users based on whether the Unit Manager has a restricted or unrestricted view.

<table>
<thead>
<tr>
<th>User Type Being Viewed</th>
<th>Amount of Detail Visible</th>
<th>Unrestricted View</th>
<th>Restricted View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Manager, Scanner or Reader in the business unit</td>
<td>Full</td>
<td>Full</td>
<td>Full</td>
</tr>
<tr>
<td>Scanner or Reader not in the business unit</td>
<td>Partial</td>
<td>Partial</td>
<td>None</td>
</tr>
<tr>
<td>Unit Manager not in the business unit</td>
<td>Partial</td>
<td>Partial</td>
<td>None</td>
</tr>
<tr>
<td>Manager</td>
<td>Partial</td>
<td>Partial</td>
<td>None</td>
</tr>
</tbody>
</table>

Full user account details include: user login, general information, assigned asset groups, user role, business unit, the Unit Manager Point of Contact (POC), the Manager POC, extended permissions and email notifications.

With a Partial view, the following details are not visible: user login, extended permissions and email notifications.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_id_contains={string}</td>
<td>(Optional) Show only user accounts with an external ID value that contains a certain string. The string you specify can have a maximum of 256 characters. The characters can be in uppercase, lowercase or mixed case (the service performs case sensitive matching). HTML or PHP tags cannot be included. Only one of these parameters may be specified for a single API request: external_id_contains or external_id_assigned.</td>
</tr>
<tr>
<td>external_id_assigned={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

**DTD**

<platform API server>/user_list_output.dtd
**Add/Edit User**

/msp/user.php

[GET]  [POST]

Add a user account or edit an existing account. You can add users to the “Unassigned” business unit or an existing, custom business unit. For each new account (except when the user role is Contact) the service automatically generates login credentials, including a login ID and “strong” password.

Permissions - Managers can add/edit user accounts in any business unit. Unit Managers can add/edit users in their own business unit. Administrators can add/edit user all accounts except Manager and Administrator user.

Express Lite - This API is available to Express Lite users. A total of 3 users can be added per subscription.

**Adding user to custom business unit**

To add users to a custom business unit, follow these steps:

- With a Manager or user administrator account, log into the Qualys user interface and create the business unit. Note business units may be created using the Qualys user interface only.

- If a Unit Manager is not already assigned to the business unit, you must add one. With a Manager account, make a user.php request to add a Unit Manager who is automatically assigned as the business unit’s point of contact (POC).

- With a Manager or Unit Manager account, make a user.php request to add other users to the custom business unit. A Manager and user administrator can add a user to any business unit, while a Unit Manager can add a user to their own business unit.

**Delivery of new account credentials to user**

When adding a new user (except Contact), the API user has the option to deliver login credentials directly to the user via email or through the application as follows.

Email notification - By default the user.php function sends the new user an email notification with a secure link to their login credentials. When the user clicks the secure link to view the credentials, the service changes the account status automatically from “Pending Activation” to “Active”.

XML output - Instead of sending an email notification, the API user has the option to return the new user’s login credentials in the XML output document. To do this, make a user.php request with the send_email=0 input parameter. As a result the service returns the user’s login ID and password as XML value pairs in the XML output, and the account status is automatically set to “Active”.

**First login completes account registration**

To complete account registration, a new user must log into the Qualys user interface with their assigned login information (platform URL and login credentials). When the user has been created using the user.php function the user can login using the Qualys user interface or using the acceptEULA.php API function. See “User Registration Process” and “Accept Qualys EULA” or more information.

**Editing accounts - edit and clear options**

For an existing account, you can edit and clear account parameters as follows.

Edit Parameters - An existing user may be edited using user.php to update the user name and general information. Additional parameters can be edited using the Qualys user interface. When editing parameters using user.php, existing parameter values are replaced with newly specified ones. For example, if you edit an existing Scanner with the assigned asset group “New York” and you wish to add the asset group “Hong Kong”, then the edit request would include `asset_groups=New+York,Hong+Kong`. An edit request can be used to clear (reset) parameters by assigning the empty string “”.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=add</td>
<td>edit</td>
</tr>
<tr>
<td>login={login}</td>
<td>(Required for Edit, not valid for Add) Specifies the Qualys user login of the user account you wish to edit. This parameter is invalid for an add request.</td>
</tr>
</tbody>
</table>

**New User - Login Credentials**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>send_email={0</td>
<td>1}</td>
</tr>
</tbody>
</table>

1 — (the default) specifies that an email notification will be sent to the new user. The user clicks a secure link in the email to view the login ID and password.

0 — specifies that an email notification will not be sent to the new user, and the XML report returned by the function will include the login ID and password for the user account as XML value pairs.
### Permissions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_role={role}</td>
<td>(Required for Add, not valid for Edit, not valid for Express Lite users) Specifies the user role. A valid value is: manager, unit_manager, scanner, reader, contact or administrator. The first user added to a new custom business unit must be unit_manager.</td>
</tr>
<tr>
<td>business_unit={title}</td>
<td>(Required for Add, not valid for Edit, not valid for Express Lite user) Specifies the user’s business unit. A valid value is “Unassigned”, or the title of an existing custom business unit. Note a custom business unit may be added using the Qualys user interface.</td>
</tr>
<tr>
<td>asset_groups={grp1,grp2...}</td>
<td>(Optional) Specifies the asset groups assigned to the user, when the user role is Scanner, Reader or Contact. Multiple asset groups are comma separated. This parameter is invalid when the user role is Manager or Unit Manager.</td>
</tr>
</tbody>
</table>

### General Information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>first_name={name}</td>
<td>(Required for Add, Optional for Edit) Specifies the user's first name. The name may include a maximum of 50 characters.</td>
</tr>
<tr>
<td>last_name={name}</td>
<td>(Required for Add, Optional for Edit) Specifies the user’s last name. The name may include a maximum of 50 characters.</td>
</tr>
<tr>
<td>title={title}</td>
<td>(Required for Add, Optional for Edit) Specifies the user’s job title. The title may include a maximum of 100 characters.</td>
</tr>
<tr>
<td>phone={value}</td>
<td>(Required for Add, Optional for Edit) Specifies the user’s phone number. This value may include a maximum of 40 characters.</td>
</tr>
<tr>
<td>fax={value}</td>
<td>(Optional) The user’s FAX number. This value may include a maximum of 40 characters.</td>
</tr>
<tr>
<td>email={value}</td>
<td>(Required for Add, Optional for Edit) Specifies the user’s email address. The address must be a properly formatted address with a maximum of 100 characters.</td>
</tr>
<tr>
<td>address1={value}</td>
<td>(Required for Add, Optional for Edit) Specifies the user’s address line 1. This value may include a maximum of 80 characters.</td>
</tr>
<tr>
<td>address2={value}</td>
<td>(Optional) Specifies the user’s address line 2. This value may include a maximum of 80 characters.</td>
</tr>
<tr>
<td>city={value}</td>
<td>(Required for Add, Optional for Edit) Specifies the user’s city. This value may include a maximum of 50 characters.</td>
</tr>
</tbody>
</table>
**Sample - Add user**

Add a new user, Chris Washington, to the Unassigned business unit with the Scanner user role, and automatically send the user an email notification with a secure link to his login credentials.

**API request:**

```plaintext
```

**XML output:**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE USER_OUTPUT SYSTEM "https://qualysapi.qualys.com/user_output.dtd">
<USER_OUTPUT>
  <API name="user.php" username="sabkl_av1" at="2018-07-20T22:54:25Z" />
</USER_OUTPUT>
```
<RETURN status="SUCCESS">
  <MESSAGE>quays_cw4 user has been successfully created.</MESSAGE>
</RETURN>
</USER_OUTPUT>

Sample - Edit user to change title

API request:

XML output:
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE USER_OUTPUT SYSTEM "https://qualysapi.qualys.com/user_output.dtd">
<USER_OUTPUT>
  <API name="user.php" username="sabkl_av1" at="2018-07-20T23:06:35Z" />
  <RETURN status="SUCCESS">
    <MESSAGE>quays_ch user has been successfully updated.</MESSAGE>
  </RETURN>
</USER_OUTPUT>

Sample - External ID

Add the external ID “Qualys123” to the existing user account “qualys_ab5” when that account does not already have an external ID:

https://qualysapi.qualys.com/msp/user.php?action=edit&login=qualys_ab5&external_id=Qualys123

Add the external ID “Qualy123” to the existing user account “qualys_ab” when that account already has an external ID:

https://qualysapi.qualys.com/msp/user.php?action=edit&login=qualys_ab5&external_id=Qualys123

Delete the external ID currently defined for the user account “qualys_ab5”:


Sample - Set Timezone

Assign a timezone to a user using the optional parameter “time_zone_code”.

Sample - Set specific timezone (i.e. pass timezone code)


Sample - Set user profile to browser's timezone (i.e. pass empty/null)

https://qualysapi.qualys.com/msp/user.php?action=edit&login=acme_ab&time_zone_code=

Looking for timezone codes? Use the time zone code list function to request the list:

<platform API server>/msp/time_zone_code_list.php

DTD

<platform API server>/user_output.dtd

Default Parameters - New User

Several user parameters are set automatically when a new user is created. These are identified below. The parameter value *** is the value defined for the user account making the API request.

<table>
<thead>
<tr>
<th>General and User Role</th>
<th>Manager</th>
<th>Unit Manager</th>
<th>Administrator</th>
<th>Scanner</th>
<th>Reader</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zip code</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
<tr>
<td>Company</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
<tr>
<td>Language - KnowledgeBase</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
</tbody>
</table>

User Status

<table>
<thead>
<tr>
<th>Pending activation</th>
<th>Pending activation</th>
<th>Pending activation</th>
<th>Pending activation</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow access to</td>
<td>GUI and API</td>
<td>GUI and API</td>
<td>GUI and API</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Notification Options

<table>
<thead>
<tr>
<th>Latest Vulnerabilities</th>
<th>Weekly</th>
<th>Weekly</th>
<th>n/a</th>
<th>Weekly</th>
<th>Weekly</th>
<th>Weekly</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan Summary</td>
<td>All</td>
<td>Scans on assigned groups</td>
<td>n/a</td>
<td>Scans on assigned groups</td>
<td>Scans on assigned groups</td>
<td>Scans on assigned groups</td>
</tr>
</tbody>
</table>
### Country codes

State codes

State Codes for United States
Value state codes when country is “United States of America”:


State Codes for Australia
Valid state codes when country is “Australia”:

No State | New South Wales | Northern Territory | Queensland | Tasmania | Victoria | Western Australia

State Codes for Canada
Valid state codes when country is “Canada”:

No State | Alberta | British Columbia | Manitoba | New Brunswick | Newfoundland | Northwest Territories | Nova Scotia | Nunavut | Ontario | Prince Edward Island | Quebec | Saskatchewan | Yukon

State Codes for India
Valid state codes when country is “India”:

No State | Andhra Pradesh | Andaman and Nicobar Islands | Arunachal Pradesh | Assam | Bihar | Chandigarh | Chattisgarh | Dadra and Nagar Haveli | Daman and Diu | Delhi | Goa | Gujarat | Haryana | Himachal Pradesh | Jammu and Kashmir | Jharkhand | Karnataka | Kerala | Lakshadweep | Madhya Pradesh | Maharashtra | Manipur | Meghalaya | Mizoram | Nagaland | Orissa | Pondicherry | Punjab | Rajasthan | Sikkim | Tamil Nadu | Tripura | Uttar Pradesh | Uttarakhand | West Bengal
User Registration Process

When a new user account is created, the service by default sends the user an email titled “Registration - Start Now”. This email includes a secure link to the user’s login information including platform URL and login credentials. Instead of sending an email notification, the API user has the option to return login credentials using user.php function with the send_email=0 input parameter.

The user must complete the first login to the service in order to complete the account registration and accept the Qualys EULA (End User License Agreement). When the first login is completed, the service sends the user an email titled “Registration - Complete”.

A new user has the option to complete the first login by simply logging into the Qualys user interface, as long as the user is granted the GUI access method. (Note a new user created using the user.php function is automatically granted the GUI and API access methods.) Using the Qualys user interface, the user is directed to the First Login form to complete the registration and accept the Qualys EULA.

The acceptEULA.php API function is provided as a programmatic method for completing the registration and accepting the Qualys EULA. To use complete the first login using the acceptEULA.php function, the user must submit an API request using their platform URL and login credentials.

Important: If a new user account is created using the Qualys user interface and the account is granted the API access method only (without the GUI access method), the user must complete the first login using the acceptEULA.php API function. If the acceptEULA.php API request is not made or it is not successful, the new account will not be activated and any API requests submitted using the new account will fail.
Accept Qualys EULA

/msp/acceptEULA.php

[GET] [POST]

Allows Qualys users to complete the registration process and accept the Qualys End User License Agreement (EULA) on behalf of their customers. This function provides programmatic acceptance of the Qualys EULA.

A new user can complete the registration process and accept the Qualys EULA through the Qualys user interface as long as their account is granted the GUI access method. (Note a new user created using the user.php function is automatically granted the GUI and API access methods.) Optionally, a new user can complete the registration and accept the Qualys EULA using the acceptEULA.php function. See User Registration Process

A Web application that allows Qualys EULA acceptance can be setup as follows. Inside the third party web application, a developer can setup a Web form that displays the Qualys EULA and has an “I Accept” button. A new Qualys user opens the Web form in a browser, reads the EULA description and clicks “I Accept” in the Web form. The third party’s program submits an HTTP request to the Qualys API server using the acceptEULA.php. Along with the acceptEULA.php URL, the application must send Qualys user account credentials (login and password) as part of the HTTP request.

Permissions - Any user with permission to log in to Qualys can complete the registration and accept the EULA.

Sample - Accept the Qualys EULA on behalf of a user

API request:

https://qualysapi.qualys.com/msp/acceptEULA.php

XML output:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE GENERIC_RETURN SYSTEM "https://qualysapi.qualys.com/generic_return.dtd">
<GENERIC_RETURN>
<API name="acceptEULA.php" username="rob" at="2018-05-10T13:44:23" />
<RETURN status="SUCCESS">
TNC accepted within MSP
</RETURN>
</GENERIC_RETURN>
```

DTD

```xml
<platform API server>/generic-return.dtd
```
Activate/Deactivate Users

/msp/user.php

[GET]  [POST]

Activate and deactivate user accounts. A user with inactivate status can be activated. A user with active status can be deactivated. Session based authentication is not supported using this API.

These actions correspond to the activate/deactivate options in the Qualys UI. Note new accounts are activated by default after the user completes the account activation process (registration) by logging into the service for the first time.

Permissions - Mangers can activate/deactivate all users in subscription. Unit Managers can activate/deactivate users in their own business unit. Administrators can activate/deactivate all users except Manager and Administrator user.

Express Lite - This API is available to Express Lite users.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=activate</td>
<td>deactivate</td>
</tr>
<tr>
<td>login={login}</td>
<td>(Required) Specifies the Qualys user login for the user account you wish to activate or deactivate.</td>
</tr>
</tbody>
</table>

Samples

Deactivate the user account “qualys_ab3” (and this account has an “Active status“):


Activate the user account “qualys_ab3” (and this account has an “Inactive” status):


DTD

<platform API server>/user_output.dtd
User Password Change

/msp/password_change.php

[GET]  [POST]

Change passwords for all or some users in the same subscription. Many Qualys customers have an internal security policy requirement to change passwords for users at a particular time interval. Changing password for multiple users at once as batch process is supported. New passwords are automatically generated by the service.

It’s possible to change passwords for user accounts with a status of “active”, “inactive” or “pending activation”. It’s not possible to change passwords for deleted accounts. Since Contact users do not have login access to Qualys, it’s not possible to change passwords for Contacts.

A password change API request returns a password change XML report indicating the user accounts affected and whether password changes were made for each account. A success message is included when passwords were changed on all target accounts. A warning message is included if passwords for any of the target accounts could not be changed. Upon error, an error message is included.

By default the password changes made by the password_change.php API causes the service to automatically send each affected user an email which notifies them of the password change. If you do not wish users to receive this email notification, you have the option to return the user login ID and password for affected users as XML value pairs in the password change report. To do this, make a password_change.php request and specify the email=0 parameter. If you make such a request on an account with the status “pending activation”, the function automatically assigns the “active” status since the login credentials are available in the XML report.

Permissions - Managers can change passwords for all users in subscription, except the user making the request. Unit Managers can change passwords for all users in same business unit, except the user making the request. Administrators can change passwords for all users in subscription, except Manager and the user making the request.

Express Lite - This API is available to Express Lite users.
Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_logins=(value)</td>
<td>(Required) Specifies one or more QualysGuard user login IDs of target user accounts. Multiple user login IDs are comma separated. Specify user_logins=all to change the password for all users in the user’s account, except the requesting user. See Permissions</td>
</tr>
<tr>
<td>email={0</td>
<td>1}</td>
</tr>
<tr>
<td></td>
<td>1 — (the default) specifies that an email notification will be sent to affected users. Each user clicks a secure link in the email to view the new password.</td>
</tr>
<tr>
<td></td>
<td>0 — specifies that email notifications will not be sent to affected users, and the XML report returned by the function will include the login ID and password for each user account as XML value pairs.</td>
</tr>
</tbody>
</table>

Samples

Make a password change request for two accounts and send affected users an email notification including a secure link to their new password:

https://qualysapi.qualys.com/msp/password_change.php?
user_logins=acme_jr,acme_dd

Make a password change request for all users in the API user’s account (except the API user) and return the login ID and password for each affected user in the password change XML response:

https://qualysapi.qualys.com/msp/password_change.php?
user_logins=all&email=0

DTD
<platform API server>/password_change_output.dtd
Export User Activity Log

/api/2.0/fo/activity_log/

[GET] [POST]

Export the user activity log for a subscription to CSV format.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required)</td>
</tr>
<tr>
<td>user_action={value}</td>
<td>(Optional) You can filter the output based on the actions. For example, login (for user login), launch (for scan launched), finished (for scan finished), etc. The actions which are included in the output depend on the user who runs the API. Managers see all actions taken by all users. Unit Managers see actions taken by users in their business unit. Scanners and Readers see their own actions only.</td>
</tr>
<tr>
<td>action_details={value}</td>
<td>(Optional) Filter on further information about the user actions. For example, for the action “error”, you can filter by the error details &quot;No connection from scanner appliance&quot;.</td>
</tr>
<tr>
<td>username={value}</td>
<td>(Optional) The name of the user who performed the action. Usernames are included in the output only if the user running the API is a Manager or a Unit Manager. A Unit Manager can see usernames only for users in the Unit Manager’s hierarchy.</td>
</tr>
<tr>
<td>since_datetime={value}</td>
<td>(Optional) Specify the date to include the activity log starting from that point in time. Date must be in the format YYYY-MM-DD HH:ii:ss, and must be less than or equal to today’s date.</td>
</tr>
<tr>
<td>until_datetime={value}</td>
<td>(Optional) Specify the date to include the activity log until a specific point in time. Date must be in the format YYYY-MM-DD HH:ii:ss, and must be more than or equal to since_datetime, and less than or equal to today’s date.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| user_role=[value] | (Optional) A Manager or Unit Manager can choose to export logs for certain user roles instead of all user roles. Specify this parameter to export logs for users with certain user roles. Multiple roles are comma separated. User roles you can specify:  
- Manager  
- Unit Manager  
- Auditor  
- Scanner  
- Reader  
- KnowledgeBase Only  
- Remediation User  
- Contact  
What logs are exported by default? For a Manager logs are exported for all users (all user roles) by default. For a Unit Manager logs are exported only for users (all user roles) in the Unit Manager’s hierarchy (i.e. business unit). |
| output_format=CSV | (Optional) CSV (default) |
| truncation_limit={value} | (Optional) Limit the number of log records to include in the CSV output. |

Sample - Export activity log to csv format

**API request:**

```bash
curl -u "username:password" -H "X-Requested-With:curl"
"https://qualysapi.qualys.com/api/2.0/fo/activity_log/?action=list"
```

**Sample CSV output:**

```
"Date","Action","Module","Details","User Name","User Role","User IP"
"2017-02-03T04:35:38Z","login","auth","user_logged in","saand_rn","Manager","10.113.195.136"
"2017-02-02T13:58:16Z","login","auth","user_logged in","saand_rn","Manager","10.113.195.136"
"2017-02-02T13:48:07Z","request","auth","API:/api/2.0/fo/activity_log/index.php","saand_rn","Manager","10.113.195.136"
"2017-02-02T13:48:07Z","request","auth","API:/api/2.0/fo/activity_log/index.php","saand_rn","Manager","10.113.195.136"
"2017-02-02T13:31:19Z","request","auth","API:/api/2.0/fo/activity_log/index.php","saand_rn","Manager","10.113.195.136"
"2017-02-02T13:28:38Z","request","auth","API:/api/2.0/fo/activity_log/index.php","saand_rn","Manager","10.113.195.136"
"2017-02-02T13:28:17Z","request","auth","API:
```

847
/api/2.0/fo/activity_log/index.php","saand_rn","Manager","10.113.195.136"  "2017-02-02T13:27:27Z","request","auth","API:
/api/2.0/fo/activity_log/index.php","saand_rn","Manager","10.113.195.136"  "2017-02-02T13:26:41Z","request","auth","API:
Appendix A - API Documentation

Looking for details on XML output and DTDs? Download this reference

Qualys API (VM, PC) XML/DTD Reference

You can find all our latest API Documentation at the Qualys Community at Qualys Documentation

HTML documentation is available through the product for your convenience. Just log into your account, choose Help > Resources from the top menu.
Appendix B - Ports used for scanning

Appendix B - Ports used for scanning
Here’s a list of ports used by Qualys Vulnerability Management to scan your host assets.
TCP Standard Scan (about 1900 ports)
TCP Light Scan (about 160 ports)
UDP Standard Scan (about 180 ports)
UDP Light Scan (about 30 ports)
TCP Standard Scan (about 1900 ports)
2109, 2111, 2115, 2120, 2140, 2160-2161, 2201-2202, 2213, 2221-2223, 2232-2239, 2241,
2260, 2279-2288, 2297, 2301, 2307, 2334, 2339, 2345, 2381, 2389, 2391, 2393-2394, 2399,
2401, 2433, 2447, 2500-2501, 2532, 2544, 2564-2565, 2583, 2592, 2600-2605, 2626-2627,
2638-2639, 2690, 2700-2702, 2716, 2766, 2784-2789, 2801, 2908-2912, 2953-2954, 2967, 2998,
3000-3002, 3006-3007, 3010-3011, 3020, 3047-3049, 3080, 3127-3128, 3141-3145, 3180-3181,
3205, 3232, 3260, 3264, 3267-3269, 3279, 3306, 3322-3325, 3333, 3340, 3351-3352, 3355,
3372, 3389, 3421, 3454-3457, 3689-3690, 3700, 3791, 3900, 3984-3986, 4000-4002, 4008-4009,
4080, 4092, 4100, 4103, 4105, 4107, 4132-4134, 4144, 4242, 4321, 4333, 4343, 4443-4454,
4500-4501, 4567, 4590, 4626, 4651, 4660-4663, 4672, 4899, 4903, 4950, 5000-5005, 50095011, 5020-5021, 5031, 5050, 5053, 5080, 5100-5101, 5145, 5150, 5190-5193, 5222, 5236,
5300-5305, 5321, 5400-5402, 5432, 5510, 5520-5521, 5530, 5540, 5550, 5554-5558, 5569,
5599-5601, 5631-5632, 5634, 5650, 5678-5679, 5713-5717, 5729, 5742, 5745, 5755, 5757,
5766-5767, 5800-5802, 5900-5902, 5977-5979, 5997-6053, 6080, 6103, 6110-6112, 6123, 6129,
6141-6149, 6253, 6346, 6387, 6389, 6400, 6455-6456, 6499-6500, 6515, 6543, 6558, 6588,
6660-6670, 6672-6673, 6699, 6767, 6771, 6776, 6789, 6831, 6883, 6912, 6939, 6969-6970,
7000-7021, 7070, 7080, 7099-7100, 7121, 7161, 7174, 7200-7201, 7300-7301, 7306-7308, 7395,
7426-7431, 7491, 7511, 7777-7778, 7781, 7789, 7895, 7938, 7999-8020, 8023, 8032, 8039,
8080-8082, 8090, 8100, 8181, 8192, 8200, 8383, 8403, 8443, 8450, 8484, 8500, 8732, 8765,
8886-8894, 8910, 9000-9002, 9005, 9043, 9080, 9090, 9098-9100, 9400, 9443, 9495, 9535,
9570, 9872-9876, 9878, 9889, 9989-10002, 10005, 10007, 10080-10082, 10101, 10202, 10204,
10520, 10607, 10666, 11000-11002, 11004, 11223, 12000-12002, 12076, 12223, 12287, 1234512346, 12361-12362, 12456, 12468-12469, 12631, 12701, 12753, 13000, 13333, 14237-14238,
15858, 16384, 16660, 16959, 16969, 17000, 17007, 17300, 18000, 18181-18186, 18190-18192,
18194, 18209-18210, 18231-18232, 18264, 19541, 20000-20001, 20011, 20034, 20200, 20203,
20331, 21544, 21554, 21845-21849, 22222, 22273, 22289, 22305, 22321, 22555, 22800, 22951,
23456, 23476-23477, 25000-25009, 25252, 25793, 25867, 26000, 26208, 26274, 26409, 2700027009, 27374, 27665, 29369, 29891, 30029, 30100-30102, 30129, 30303, 30999, 31336-31337,
31339, 31554, 31666, 31785, 31787-31788, 32000, 32768-32790, 33333, 33567-33568, 33911,

850


Appendix B - Ports used for scanning

34324, 37651, 40412, 40421-40423, 42424, 44337, 47557, 47806, 47808, 49400, 50000-50001, 50505, 50766, 51102, 51107, 51112, 53001, 54320-54321, 57341, 60008, 61439, 61466, 62078, 65000, 65301, 65512

TCP Light Scan (about 160 ports)

UDP Standard Scan (about 180 ports)

UDP Light Scan (about 30 ports)
7, 13, 17, 19, 37, 53, 67-69, 111, 123, 135, 137, 161, 177, 407, 464, 500, 517-518, 520, 1434, 1645, 1701, 1807, 2049, 3527, 4569, 4665, 5036, 5060, 5632, 6502, 7778, 15345
Appendix C - Scan Results JSON

This section describes all the possible keys involved when a Scan API “fetch” request is made in JSON format (/api/2.0/fo/scan/?action-fetch&output_format=json). Click here for sample JSON output

A list of keys for various scan scenarios is provided

Scan Finished with Vulnerabilities
Scan Cancelled
Scan Error
Scan Finished (Host Not Alive)
Scan Paused
Scan Interrupted

Scan Finished with Vulnerabilities

Scan Job
launch_date, active_hosts, total_hosts, type, status, reference, scanner_appliance, duration, scan_title, asset_groups, ips, excluded_ips, option_profile

Per Host
ip, dns, netbios, os, ip_status, qid, title, type, severity, port, protocol, fqdn, ssl, cve_id, vendor_reference, bugtraq_id, cvss_base, cvss_temporal, cvss3_base, cvss3_temporal, threat, impact, solution, exploitability, associated_malware, results, pci_vuln, instance, os_cpe, category, instance

If PCI is Enabled
pci_vuln

Host Stats

target_distribution_across_scanner_appliances
hosts_not_scanned_excluded_host_ip
hosts_not_scanned_host_not_alive_ip
hosts_not_scanned_host_not_alive_dns
hosts_not_scanned_host_not_alive_netbios
hosts_not_scanned_hostname_not_found_ip
hosts_not_scanned_scan_discontinued_ip
hosts_not_scanned_scan_discontinued_netbios_instance_ids
hosts_not_scanned_scan_discontinued_netbios_dns
hosts_not_scanned_scan_discontinued_netbios
hosts_not_scanned_dns_hostname_could_not_be_resolved
hosts_not_scanned_dns_netbios_could_not_be_resolved

hosts_not_scanned_dns_could_not_be_resolved
hosts_not_scanned_ip_could_not_be_resolved
Appendix C - Scan Results JSON

hosts_not_scanned_hostname_not_found_netbios
hosts_not_scanned_hostname_not_found_dns

Scan Cancelled

Scan Job

launch_date, active_hosts, total_hosts, type, status, reference, scanner_appliance, duration, scan_title, asset_groups, ips, excluded_ips, option_profile

Host Stats

no_vulnerabilities_match_your_filters_for_these_hosts

host_not_scanned, scan_canceled_by_user_ip_
host_not_scanned, scan_canceled_by_administrator_ip_
host_not_scanned, scan_canceled_by_service_ip_
host_not_scanned, scan_canceled_by_unknown_ip_

host_not_scanned, scan_canceled_by_user, (#No of IP) hosts
host_not_scanned, scan_canceled_by_administrator, (#No of IP) hosts
host_not_scanned, scan_canceled_by_service, (#No of IP) hosts
host_not_scanned, scan_canceled_by_unknown, (#No of IP) hosts

host_not_scanned, scan_canceled_by_user_dns_
host_not_scanned, scan_canceled_by_administrator_dns_
host_not_scanned, scan_canceled_by_service_dns_
host_not_scanned, scan_canceled_by_unknown_dns_

host_not_scanned, scan_canceled_by_user_instance_ids_
host_not_scanned, scan_canceled_by_administrator_instance_ids_
host_not_scanned, scan_canceled_by_service_instance_ids_
host_not_scanned, scan_canceled_by_unknown_instance_ids_

host_not_scanned, scan_canceled_by_user, dns,(#No of DNS) hosts
host_not_scanned, scan_canceled_by_administrator, dns,(#No of DNS) hosts
host_not_scanned, scan_canceled_by_service, dns,(#No of DNS) hosts
host_not_scanned, scan_canceled_by_unknown, dns,(#No of DNS) hosts

host_not_scanned, scan_canceled_by_user, instance_ids,(#No of DNS) hosts
host_not_scanned, scan_canceled_by_administrator, instance_ids,(#No of DNS) hosts
host_not_scanned, scan_canceled_by_service, instance_ids,(#No of DNS) hosts
host_not_scanned, scan_canceled_by_unknown, instance_ids,(#No of DNS) hosts

host_not_scanned, scan_canceled_by_user_netbios_
host_not_scanned, scan_canceled_by_administrator_netbios_
host_not_scanned, scan_canceled_by_service_netbios_
host_not_scanned, scan_canceled_by_unknown_netbios_

host_not_scanned, scan_canceled_by_user_netbios,(#No of Netbios) hosts
host_not_scanned, scan_canceled_by_administrator_netbios, (#No of Netbios) hosts
host_not_scanned, scan_canceled_by_service_netbios, (#No of Netbios) hosts
host_not_scanned, scan_canceled_by_unknown_netbios, (#No of Netbios) hosts

853
Appendix C - Scan Results JSON

host_not_scanned, _scan_canceled_by_unknown, netbios, (#No of Netbios) hosts

Scan Error
Scan Job
launch_date, active_hosts, total_hosts, type, status, reference, scanner_appliance, duration, scan_title, asset_groups, ips, excluded_ips, option_profile

Host Stats
no_vulnerabilities_match_your_filters_for_these_hosts

Scan Finished (Host Not Alive)
Scan Job
launch_date, active_hosts, total_hosts, type, status, reference, scanner_appliance, duration, scan_title, asset_groups, ips, excluded_ips, option_profile

Host Stats
target_distribution_across_scanner_appliances
hosts_not_scanned_host_not_alive_ip

Scan Paused
Scan Job
launch_date, active_hosts, total_hosts, type, status, reference, scanner_appliance, duration, scan_title, asset_groups, ips, excluded_ips, option_profile, network

Per Host
ip, dns, netbios, os, ip_status, qid, title, type, severity, port, protocol, fqdn, ssl, cve_id, vendor_reference, bugtraq_id, cvss_base, cvss_temporal, cvss3_base, cvss3_temporal, threat, impact, solution, exploitability, associated_malware, results, pci_vuln, instance, os_cpe, category

Host Stats
target_distribution_across_scanner_appliances
hosts_not_scanned_host_not_alive_ip
host_not_scanned, _scan_paused_by_service_ip_
no_vulnerabilities_match_your_filters_for_these_hosts

host_not_scanned, _scan_paused_by_user_ip_
host_not_scanned, _scan_paused_by_administrator_ip_
host_not_scanned, _scan_paused_by_service_ip_
host_not_scanned, _scan_paused_by_unknown_ip_

host_not_scanned, _scan_paused_by_user, (#No of IP) hosts
host_not_scanned, _scan_paused_by_administrator, (#No of IP) hosts
Appendix C - Scan Results JSON

host_not_scanned, scan_paused_by_service, (#No of IP) hosts
host_not_scanned, scan_paused_by_unknown, (#No of IP) hosts

host_not_scanned, scan_paused_by_user_dns
host_not_scanned, scan_paused_by_administrator_dns
host_not_scanned, scan_paused_by_service_dns
host_not_scanned, scan_paused_by_unknown_dns

host_not_scanned, scan_paused_by_user_instance_ids
host_not_scanned, scan_paused_by_administrator_instance_ids
host_not_scanned, scan_paused_by_service_instance_ids
host_not_scanned, scan_paused_by_unknown_instance_ids

host_not_scanned, scan_paused_by_user, dns, (#No of DNS) hosts
host_not_scanned, scan_paused_by_administrator, dns, (#No of DNS) hosts
host_not_scanned, scan_paused_by_service, dns, (#No of DNS) hosts
host_not_scanned, scan_paused_by_unknown, dns, (#No of DNS) hosts

host_not_scanned, scan_paused_by_user, instance_ids, (#No of DNS) hosts
host_not_scanned, scan_paused_by_administrator, instance_ids, (#No of DNS) hosts
host_not_scanned, scan_paused_by_service, instance_ids, (#No of DNS) hosts
host_not_scanned, scan_paused_by_unknown, instance_ids, (#No of DNS) hosts

host_not_scanned, scan_paused_by_user, netbios, (#No of Netbios) hosts
host_not_scanned, scan_paused_by_administrator, netbios, (#No of Netbios) hosts
host_not_scanned, scan_paused_by_service, netbios, (#No of Netbios) hosts
host_not_scanned, scan_paused_by_unknown, netbios, (#No of Netbios) hosts

host_not_scanned, scan_paused_by_user_netbios
host_not_scanned, scan_paused_by_administrator_netbios
host_not_scanned, scan_paused_by_service_netbios
host_not_scanned, scan_paused_by_unknown_netbios

Scan Interrupted

Scan Job
launch_date, active_hosts, total_hosts, type, status, reference,
scanner_appliance, duration, scan_title, asset_groups, ips, excluded_ips,
option_profile, network

Host Stats
no_vulnerabilities_match_your_filters_for_these_hosts

host_not_scanned, scan_unknown_by_user_ip
host_not_scanned, scan_unknown_by_administrator_ip
host_not_scanned, scan_unknown_by_service_ip
host_not_scanned, scan_unknown_by_unknown_ip

host_not_scanned, scan_unknown_by_user_dns
host_not_scanned, scan_unknown_by_administrator_dns
host_not_scanned, scan_unknown_by_service_dns
host_not_scanned, scan_unknown_by_unknown_dns
host_not_scanned, scan_unknown_by_user_instance_ids
host_not_scanned, scan_unknown_by_administrator_instance_ids
host_not_scanned, scan_unknown_by_service_instance_ids
host_not_scanned, scan_unknown_by_unknown_instance_ids

host_not_scanned, scan_unknown_by_user, (#No of IP) hosts
host_not_scanned, scan_unknown_by_administrator, (#No of IP) hosts
host_not_scanned, scan_unknown_by_service, (#No of IP) hosts
host_not_scanned, scan_unknown_by_unknown, (#No of IP) hosts

host_not_scanned, scan_unknown_by_user, dns, (#No of DNS) hosts
host_not_scanned, scan_unknown_by_administrator, dns, (#No of DNS) hosts
host_not_scanned, scan_unknown_by_service, dns, (#No of DNS) hosts
host_not_scanned, scan_unknown_by_unknown, dns, (#No of DNS) hosts

host_not_scanned, scan_unknown_by_user, instance_ids, (#No of DNS) hosts
host_not_scanned, scan_unknown_by_administrator, instance_ids, (#No of DNS) hosts
host_not_scanned, scan_unknown_by_service, instance_ids, (#No of DNS) hosts
host_not_scanned, scan_unknown_by_unknown, instance_ids, (#No of DNS) hosts

host_not_scanned, scan_unknown_by_user_netbios
host_not_scanned, scan_unknown_by_administrator_netbios
host_not_scanned, scan_unknown_by_service_netbios
host_not_scanned, scan_unknown_by_unknown_netbios

host_not_scanned, scan_unknown_by_user, netbios, (#No of Netbios) hosts
host_not_scanned, scan_unknown_by_administrator, netbios, (#No of Netbios) hosts
host_not_scanned, scan_unknown_by_service, netbios, (#No of Netbios) hosts
host_not_scanned, scan_unknown_by_unknown, netbios, (#No of Netbios) hosts

hosts not scanned, hostname not found, (#NumberOfNoTrackerIP) hosts
hosts not scanned, hostname not found, netbios, (#NumberOfNoTrackerNETBIOS) hosts
hosts not scanned, hostname not found, dns, (#NumberOfNoTrackerDNS) hosts
hosts not scanned, hostname not found, instance ids, (#NumberOfNoTrackerInstanceIDs) hosts

hosts_not_scanned_excluded_host_dns
hosts_not_scanned_excluded_host_instance_ids
hosts_not_scanned_host_not_alive_dns
hosts_not_scanned_host_not_alive_instance_ids

Sample JSON output

```json
[
  {
    "scan_report_template_title": "Scan Results",
    "result_date": "06/29/2018 06:19:26",
    "company": "Qualys, Inc",
    "add1": "919 E Hillsdale Blvd, 4th Floor",
    "add2": null,
  }
]```

856
"city": "Foster City",
"state": "California",
"country": "United States of America",
"zip": "94404",
"name": "Mayur Mistry",
"username": "mayur_mmm",
"role": "Manager"
},
{
  "scan_date": "09/29/2018 21:20:35",
  "active_hosts": null,
  "total_hosts": "457660",
  "type": "On Demand",
  "status": "Canceled",
  "reference": "scan/1527628838.16797",
  "scanner_appliance": "",
  "duration": "00:00:24",
  "scan_title": "My Scan",
  "asset_groups": "4.5LIPS",
  "ips": "10.10.0.0, 10.10.0.2, 10.10.0.4, 10.10.0.6",
  "excluded_ips": "",
  "option_profile": "Initial Options"
},
{
  "host_not_scanned, scan_canceled_by_user_ip_": "10.10.0.0, 10.10.0.2, 10.10.0.4, 10.10.0.6"
}
# Appendix D - Error Codes / Descriptions

Here’s a list of Qualys API error codes along with a description of what each code means. For an API request that had an error, you’ll find the error code and text in the XML response.

<table>
<thead>
<tr>
<th>HTTP Status</th>
<th>Error Code</th>
<th>Error Text</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP/1.1</td>
<td>1901</td>
<td>Unrecognized parameter(s):...</td>
<td>The API request contained one or more parameters which are not supported, or are not available to the browsing user.</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>1903</td>
<td>Missing required parameter(s):...</td>
<td>The API request did not contain one or more parameters which are required.</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>1904</td>
<td>Please specify only one of these parameters:...</td>
<td>The API request contained 2 or more parameters from a group from which at most one may be specified.</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>1905</td>
<td>parameter ... has invalid value ...</td>
<td>The API request contained a valid parameter specified with an invalid value.</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>1907</td>
<td>The following combination of key=value pairs is not supported:...</td>
<td>The API request contained an invalid or unsupported combination of parameters.</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>1908</td>
<td>Request method (GET or POST) is incompatible with specified parameter(s):...</td>
<td>The API request was made with an unsupported HTTP request method (GET or POST or PUT or DELETE or HEAD).</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>1920</td>
<td>The requested operation is blocked by one or more existing Business Objects</td>
<td>The API request was blocked by other API requests. In practice this should be replaced by one of error code 1960 or 1965 (see below).</td>
</tr>
<tr>
<td>409 Conflict</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>1960</td>
<td>The requested operation is blocked by one or more existing Business Objects</td>
<td>Too many other API requests currently running (i.e. concurrency limit).</td>
</tr>
<tr>
<td>409 Conflict</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>1965</td>
<td>The requested operation is blocked by one or more existing Business Objects</td>
<td>Too many other API requests have run recently (i.e. rate limit).</td>
</tr>
<tr>
<td>409 Conflict</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>1922</td>
<td>Please specify at least one of the following parameters:...</td>
<td>The API request was missing some required information (but not necessarily a single specific parameter).</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>1981</td>
<td>Your request is being processed. Please try this same request again later.</td>
<td>The API request is for a business operation which is already underway.</td>
</tr>
<tr>
<td>202 Accepted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP Status</td>
<td>Error Code</td>
<td>Error Text</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------</td>
<td>-------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HTTP/1.1 400 Bad Request</td>
<td>999</td>
<td>Internal Error</td>
<td>The API request failed for some reason having to do with the (client) request. In practice this should always be expressed as some other error type, giving more information about what was actually wrong with the request.</td>
</tr>
<tr>
<td>HTTP/1.1 501 Internal Error</td>
<td>999</td>
<td>Internal Error</td>
<td>The API request failed due to a problem with QWEB.</td>
</tr>
<tr>
<td>HTTP/1.1 503 Maintenance</td>
<td>1999</td>
<td>We are performing scheduled maintenance on our System. We apologize for any inconvenience.</td>
<td>The API request failed because the Qualys Cloud Platform is in maintenance mode.</td>
</tr>
<tr>
<td>HTTP/1.1 401 Unauthorized</td>
<td>2000</td>
<td>Bad Login/Password</td>
<td>The API request failed because of an authentication failure.</td>
</tr>
<tr>
<td>HTTP/1.1 403 Forbidden</td>
<td>2002</td>
<td>User account is inactive.</td>
<td>The API request failed because of an authorization failure.</td>
</tr>
<tr>
<td>HTTP/1.1 409 Conflict</td>
<td>2003</td>
<td>Registration must be completed before API requests will be served for this account</td>
<td>The API request failed because nobody has yet accepted the EULA on behalf of the user’s subscription.</td>
</tr>
<tr>
<td>HTTP/1.1 409 Conflict</td>
<td>2011</td>
<td>SecureID authentication is required for this account, so API access is blocked</td>
<td>The API request failed because SecureID authentication won’t work with API calls.</td>
</tr>
<tr>
<td>HTTP/1.1 403 Forbidden</td>
<td>2012</td>
<td>User license is not authorized to run this API.</td>
<td>The API request failed because the user’s subscription does not have API access enabled.</td>
</tr>
</tbody>
</table>
# PCRS Error Codes

This section includes information on the PCRS error codes that you may encounter and their descriptions.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Action Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Code: 401</td>
<td>Incorrect credentials provided by the user.</td>
<td>Enter correct credentials and run the API again.</td>
</tr>
<tr>
<td></td>
<td>{ &quot;authentication_exceptions&quot;: [ &quot;Authentication Failure: 1 errors, 0 successes:authenticationFailure.e.InvalidCredentialsException&quot; ] }</td>
<td></td>
</tr>
<tr>
<td>Error code: 401</td>
<td>Expired authentication token entered by user.</td>
<td>Enter correct valid token from /auth API and call the API again.</td>
</tr>
<tr>
<td></td>
<td>{ &quot;status&quot;: 401, &quot;error&quot;: &quot;Unauthorized&quot;, &quot;message&quot;: &quot;JWT expired&quot; }</td>
<td></td>
</tr>
<tr>
<td>Error code: 401</td>
<td>Invalid token provided.</td>
<td>Enter correct valid token from /auth API and call the APIs again.</td>
</tr>
<tr>
<td></td>
<td>{ &quot;status&quot;: 401, &quot;error&quot;: &quot;Unauthorized&quot;, &quot;message&quot;: &quot;Not authenticated&quot; }</td>
<td></td>
</tr>
<tr>
<td>200 ok</td>
<td>Compliance Policy does not exist</td>
<td>Enter correct policy ID and run the API again.</td>
</tr>
<tr>
<td></td>
<td>{ &quot;policyId&quot;: &quot;94473&quot;, &quot;subscriptionId&quot;: &quot;1981983&quot;, &quot;hostIds&quot;: [], &quot;error&quot;: &quot;Error: Compliance Policy does not exist.&quot; }</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>Policy ID entered contains non-numeric characters.</td>
<td>Enter only numeric value for policy IDs.</td>
</tr>
<tr>
<td></td>
<td>[&quot;response&quot;:{&quot;datetime&quot;:&quot;2021-10-28T13:29:06Z&quot;,&quot;code&quot;:1905,&quot;text&quot;:&quot;parameter policy_ids has invalid value: 4199751a (Invalid value: 4199751a)&quot;}]</td>
<td></td>
</tr>
</tbody>
</table>
### Error Codes / Descriptions

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Action Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>404</td>
<td>Invalid parameter for &quot;evidenceRequired&quot;.</td>
<td>Enter integer value - 0 or 1 for the evidenceRequired request parameter.</td>
</tr>
<tr>
<td>404</td>
<td>Invalid input parameter for &quot;compressionRequired&quot;</td>
<td>Enter integer value - 0 or 1 for the compressionRequired request parameter.</td>
</tr>
<tr>
<td></td>
<td>Connection broken (client error)</td>
<td>Output stream has no data in stream.</td>
</tr>
<tr>
<td></td>
<td>Server side error</td>
<td>Client and server connection is broken due to connectivity loss then below error will get on server logs.</td>
</tr>
<tr>
<td>403</td>
<td>This error occurs in case of an SCA-only account.</td>
<td>Use a valid PC license.</td>
</tr>
</tbody>
</table>

You can find all our latest API Documentation at the Qualys Community at [Qualys Documentation](https://www.qualys.com/community/).
## Appendix E - Streaming Posture API Client Sample Code (Python)

The following table contains output parameters and their descriptions:

Following is a sample code to demonstrate how to use Qualys Policy Compliance Streaming Posture API to download host posture by using Python script.

<table>
<thead>
<tr>
<th>Output Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Posture record ID</td>
</tr>
<tr>
<td>instance</td>
<td>Instance type/name</td>
</tr>
<tr>
<td>policyId</td>
<td>Policy ID</td>
</tr>
<tr>
<td>controlId</td>
<td>Control ID</td>
</tr>
<tr>
<td>technologyId</td>
<td>Technology ID</td>
</tr>
<tr>
<td>status</td>
<td>Posture Status</td>
</tr>
<tr>
<td>previousStatus</td>
<td>Previous Posture Status</td>
</tr>
<tr>
<td>firstFailDate</td>
<td>Posture first fail date</td>
</tr>
<tr>
<td>lastFailDate</td>
<td>Posture last fail date</td>
</tr>
<tr>
<td>firstPassDate</td>
<td>Posture pass date</td>
</tr>
<tr>
<td>lastPassDate</td>
<td>Posture last date</td>
</tr>
<tr>
<td>postureModifiedDate</td>
<td>Posture last modified date</td>
</tr>
<tr>
<td>lastEvaluatedDate</td>
<td>Posture last evaluated date</td>
</tr>
<tr>
<td>created</td>
<td>Posture creation date</td>
</tr>
<tr>
<td>hostId</td>
<td>Host ID</td>
</tr>
<tr>
<td>ip</td>
<td>Asset instance IP address</td>
</tr>
<tr>
<td>trackingMethod</td>
<td>Asset tracking method</td>
</tr>
<tr>
<td>os</td>
<td>Asset instance operating system</td>
</tr>
<tr>
<td>osCpe</td>
<td>OS Platform Enumeration</td>
</tr>
<tr>
<td>dns</td>
<td>Host ID</td>
</tr>
<tr>
<td>qgHostid</td>
<td>QualysGuard Host ID</td>
</tr>
<tr>
<td>networkId</td>
<td>Network ID</td>
</tr>
<tr>
<td>networkName</td>
<td>Network name</td>
</tr>
<tr>
<td>complianceLastScanDate</td>
<td>Policy Compliance last scan date</td>
</tr>
<tr>
<td>customerUuid</td>
<td>Customer UUID</td>
</tr>
<tr>
<td>customerId</td>
<td>Customer ID</td>
</tr>
<tr>
<td>assetId</td>
<td>Asset ID</td>
</tr>
<tr>
<td>technology : id</td>
<td>Technology ID</td>
</tr>
<tr>
<td>technology : name</td>
<td>Technology name</td>
</tr>
<tr>
<td>criticality : label</td>
<td>Control criticality label</td>
</tr>
</tbody>
</table>
You need to install requests library such as PIP Install Requests.

```python
import requests
from requests.exceptions import Timeout
import json
import datetime
import time
import sys
import zlib

# Function to handle various errors
def handlerError(size, error):
    print('Total size downloaded %.2fm' % (size / 1048576))
    # Print total data downloaded in MBs
    print(type(error))
    print(error.args)
    print(error)

# First authenticate the user to get the token needed for subsequent API calls
headers = {'Content-Type': 'application/x-www-form-urlencoded'}
authUrl = 'https://gateway.<assigned URL>/auth'  # Replace username and password with actual userid and password. For token, the value can be 'true' or 'True'.
data = {'username': 'username', 'password': 'password', 'token': 'true'}
authResp = requests.post(authUrl, data=data, headers=headers, verify=False)
token = authResp.content.decode('utf-8')

# Use the token returned by the authentication call
# Retrieve the host IDs associated with the particular policy
headers = {
    'accept': 'application/json',
    'Authorization': 'Bearer ' + token
}
params = {'policyId': 'policyid'}
# Replace with the actual policy ID, pass multiple policy IDs as
```
Appendix E - Streaming Posture API Client Sample Code (Python)

```
comma-separated list

url='https://gateway.<assigned URL>/pcrs/1.0/posture/hostids'
response=requests.get(url, params=params, headers=headers, verify=False)

# check the response of host IDs API
if (response.status_code!=200):
    print("Unexpected response from hostids API: ")
    print(response.status_code)
    exit()

# Pass the host IDs retrieved in the previous APIs to posture info API
headers={
    'accept':'application/json',
    'Authorization': 'Bearer '+token,
    'Content-Type':'application/json'}

postureUrl = 'https://gateway.<assigned URL>/pcrs/1.0/posture/postureInfo?evidenceRequired=1&compressionRequired=1'
# If compression is used and you want to decompress the data on the fly
# with open("output.json", 'wb') as f: #If compression is used the zip file is to be stored
with open("output.json", 'w') as f: #If compression is used and decompressing on the fly or no compression used
    print('API Invoked at: '
    print(datetime.datetime.now())
    size=0.0
    try:
        with requests.post(url=postureUrl, headers=headers, data=response.content, stream=True, timeout=3600, verify=False) as postureStream:
            if (postureStream.status_code!=200):
                print("Unexpected response from posture API: ")
                print(postureStream.status_code)
                exit()
            print('First response received at: ')
            start = time.time()
            print(datetime.datetime.now())
            for chunk in postureStream.iter_content(chunk_size=1048576):
```
Appendix E - Streaming Posture API Client Sample Code (Python)

```python
if chunk:
    chunk_size = len(chunk)
    end = time.time() + 1
    outstr = d.decompress(chunk)  # If compression is not used or storing zip file, please comment this line
    size += len(outstr)
    print('Download speed: [%.2fkbps], Chunk size: [%.2fk], total size: [%.2fm] at time %s' % ((size/(end-start))/1024,
        chunk_size/1024,size/1048576,datetime.datetime.now().strftime("%H:%M:%S")),end="\r")
    f.write(outstr.decode())
    f.flush()
postureStream.close
f.flush()
print('
API finished at')
print(datetime.datetime.now())
except Exception as e:
    handlerError(size, e)
except ProtocolError as pe:
    handlerError(size, pe)
f.close()
```

# Following is a sample code to demonstrate how to use Qualys Policy Compliance Streaming Posture API for concurrent processing of host posture by using Python script.

# You need to install requests library such as PIP install requests, PIP install json_stream, and PIP install dicttoxml.

```python
import requests
from requests.exceptions import Timeout
import json
import datetime
import time
import sys
import zlib
import threading
import json_stream

def worker():
    with open("output.json", 'r') as f:
        data = json_stream.load(f)
        count = 0
        for posture in data.persistent():
            print('Count [%d] Control Id [%s] IP[%s] Criticality [%s] Status[%s]' % (count,
```
Appendix E - Streaming Posture API Client Sample Code (Python)

posture['controlId'],
posture['ip'],
posture['criticality']['label'],posture['status'],end="\r")
    count += 1

f.close()

# Function to handle various errors
def handlerError(size, error):
    print(type(error))
    print(error.args)
    print(error)

# First authenticate the user to get the token needed for subsequent API calls
headers = {'Content-Type': 'application/x-www-form-urlencoded'}
authUrl='https://gateway.<assigned URL>/auth'  #
data = {'username': <USER NAME>,
    'password':<PASSWORD>,'token':true}  # Replace username and password with actual user ID and password. For token, the value can be 'true' or 'True'.
authResp=requests.post(authUrl, data=data, headers=headers, verify=False)
token=authResp.content.decode('utf-8')

# Use the token returned by the authentication call
# Retrieve the host ids associated with the particular policy
headers={
    'accept':'application/json',
    'Authorization': 'Bearer '+token}

params={'policyId':'xxx'}
#Replace with the policyid, pass multiple policyids as comma separated list

url='https://gateway.<assigned URL>/pcrs/1.0/posture/hostids'
response=requests.get(url, params=params, headers=headers, verify=False)

# check the response of host ids API
if (response.status_code!=200):
    print("Unexpected response from host ids API: ")
    print(response.status_code)
    exit()

# Pass the host ids retrieved in the previous APIs to posture
info API
headers={
    'accept': 'application/json',
    'Authorization': 'Bearer ' + token,
    'Content-Type': 'application/json'}

postureUrl = 'https://gateway.<assigned
URL>/pcrs/1.0/posture/postureInfo?evidenceRequired=0&compressio
nRequired=1'
# If compression is used and you want to decompress the data on
the fly
d = zlib.decompressobj(16+zlib.MAX_WBITS)

apiTime = datetime.datetime.now()
with open("output.json", 'w') as f:  # If compression is used and
decompressing on the fly or no compression used
    print('API Invoked at: ')
    print(apiTime)
    t1 = threading.Thread(target=worker, daemon=True)
try:
    with requests.post(url=postureUrl, headers=headers,
data=response.content, stream=True, timeout=3600, verify=False)
    as postureStream:
        if (postureStream.status_code!=200):
            print("Unexpected response from posture API: ")
            print(postureStream.status_code)
            exit()
        print('First response received at: ')
        start = time.time()
        print(apiTime)
        count=0
        for chunk in
            postureStream.iter_content(chunk_size=1048576):
                if chunk:
                    outstr = d.decompress(chunk)  # If compression
is not used or storing zip file, please comment this line
                    f.write(outstr.decode())
                    f.flush()
                    if count == 0:
                        # turn-on the worker thread
                        t1.start()
                    count += 1

postureStream.close
apiTime = datetime.datetime.now()
except Exception as e:
    handlerError(size, e)
except ProtocolError as pe:
    handlerError(size, pe)
f.close()
t1.join()
print('\nAPI Finished at [%s] All the processing completed at [%s]'%(apiTime,datetime.datetime.now()))
Index

A
add IP addresses 499
API limits 13
Application Server authentication 256
asset search report 628
authentication 10, 247
authentication to your account 18

B
basic HTTP authentication 18

C
cancel report 619
characters in URLs 12
compliance control list 699
compliance policy export 729
compliance policy import 749
compliance policy list 725
compliance policy merge 751
compliance policy, manage asset groups 761
compliance scan list 42
country codes 839
Cyberscope report 822

date format 12
delete report 626
discovery scans 116
Docker authentication 279, 305
download report 620

E
Expires header 22

G
GET method 12

H
header parameter 18
host detection list 522
host list 505
HTTP authentication 282
HTTP Expires header 22

I
IBM DB2 authentication 285
invalid tickets 681
IP list 496, 787
IPv6 asset mapping record list 587

L
launch compliance scan 48
launch compliance scan on EC2 assets 49
launch report 599
launch scorecard report 612
launch VM scan 33
launch VM scan on EC2 assets 35

M
manage compliance scans 51
manage VM scans 38
maps 116
MariaDB authentication 308, 332
MongoDB authentication 322
MS SQL authentication 337
MySQL authentication 346

N
network
assign appliance to 594
create 591
update 593
network list 590
network maps 116
network support 590

O

option profiles
  export 147
    for compliance 194
    for PCI 186
    for VM 170
  import 160
Oracle authentication 362
Oracle Listener authentication 370
Oracle WebLogic authentication 372
overdue tickets 680

P

Palo Alto Firewall authentication 375
POST method 12
PostgreSQL authentication 386

Q

Qualys
  user account 10
Qualys End User Agreement (EULA) 842
Qualys EULA 842
Qualys Support 9
Qualys user account 10

R

report
  asset search 628
  cancel 619
  Cyberscope report 822
  delete 626
  download 620
  SCAP ARF report 826
  scorecard report 612
report DTDs, most recent 13
report, launch 599
reports
  date format 12
  decoding reports 13

S

scan authentication
  Application Server 256
  Docker 279, 305
  HTTP 282
  IBM DB2 285
  MariaDB 308, 332
  MongoDB 322
  MS SQL 337
  MySQL 346
  Oracle 362
  Oracle Listener 370
  Oracle WebLogic 372
  Palo Alto Firewall 375
  PostgreSQL 386
  SNMP 403
  Sybase 408
  Unix 414
  VMware 438
  Windows 444, 452
scan list parameters 78
scan schedules 57
scanner appliances
  list 126
    manage virtual 131
    replace 140
    update physical 137
SCAP ARF report 826
SCAP Cyberscope Report 822
SCAP policy list 827
SCAP scan list 46
scheduled report, launch 628
scheduled reports list 627
session based authentication 19
session login 22
session logout 24
session timeout 22
share PCI scan 112
SNMP authentication 403
special characters in URLs 12
state codes
  Australia  840
  Canada  840
  India  840
  United States of America  840
Sybase authentication  408

T

ticket state/status  680

U

Unix authentication  414
updated IP addresses  501
URL elements  13
URL encoded variables  12
user account
  login credentials  10
user management functions
    acceptEULA.php  842
user.php function
  country codes  839
  state codes  840
UTF-8 encoding  12

V

VM scan list  27
VM scan statistics  93
VM scan summary (hosts not scanned)  107
VM scans  27
VMware authentication  438

W

Windows authentication  444, 452