



Qualys Cloud Platform v2.x

API Release Notes

Version 2.42

November 14, 2019

Qualys Cloud Suite API gives you many ways to integrate your programs and API calls with Qualys capabilities. You'll find all the details in our user guides, available at the time of release. Just log in to your Qualys account and go to Help > Resources.

What's New

[AWS Asset Data Connector: Support for new regions](#)

[Host Asset API: Search IBM assets](#)

[SAQ - New Campaign API](#)

[WAS: Tag Details in Web App API](#)

[WAS: XSS Payloads Option for Standard Scans](#)

[WAS: New Groups for Information Gathered Issues](#)

[WAS: Cancel Scan with Results Support for Scans](#)

[WAS: Scan Again Support for Scan API](#)

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.

AWS Asset Data Connector: Support for new regions

API affected	/qps/rest/2.0/get/am/awsassetdataconnector/<id> /qps/rest/2.0/create/am/awsassetdataconnector /qps/rest/2.0/update/am/awsassetdataconnector /qps/rest/2.0/search/am/awsassetdataconnector
New or Updated APIs	Updated
DTD or XSD changes	No

The Asset Management and Tagging API has been updated to support the following new regions:

GovCloud: us-gov-east-1

China: cn-northwest-1

Bahrain: me-south-1

Sample 1 : Create new AWS asset data connector for GovCloud (us-gov-east-1)

API request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/create/am/awsassetdataconnector" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>gov-cloud</name>
      <activation>
        <set>
          <ActivationModule>VM</ActivationModule>
          <ActivationModule>PC</ActivationModule>
        </set>
      </activation>
      <isGovCloudConfigured>true</isGovCloudConfigured>
      <authRecord>
        <id>134601</id>
      </authRecord>
      <allRegions>false</allRegions>
      <disabled>false</disabled>
```

```
    </AwsAssetDataConnector>  
  </data>  
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/a  
m/awsassetdataconnector.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <AwsAssetDataConnector>  
      <id>149008</id>  
      <name>gov-cloud</name>  
      <awsAccountId>205767712438</awsAccountId>  
      <connectorState>PENDING</connectorState>  
      <type>AWS</type>  
      <defaultTags>  
        <list/>  
      </defaultTags>  
      <activation>  
        <list>  
          <ActivationModule>VM</ActivationModule>  
          <ActivationModule>PC</ActivationModule>  
        </list>  
      </activation>  
      <disabled>false</disabled>  
      <isGovCloudConfigured>true</isGovCloudConfigured>  
      <authRecord/>  
      <endpoints>  
        <list>  
          <AwsEndpointSimple>  
            <regionCode>us-gov-west-1</regionCode>  
            <regionCode>us-gov-east-1</regionCode>  
          </AwsEndpointSimple>  
        </list>  
      </endpoints>  
      <allRegions>false</allRegions>  
    </AwsAssetDataConnector>  
  </data>  
</ServiceResponse>
```

Sample 2 : Create new AWS asset data connector for China (cn-northwest-1)

Provide the new region under `AwsEndpointSimple.regionCode` in the request POST data.

API request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/create/am/awsassetdata
connector" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>cn-conn1</name>
      <authRecord>
        <id>132601</id>
      </authRecord>
      <endpoints>
        <add>
          <AwsEndpointSimple>
            <regionCode>cn-north-1</regionCode>
          </AwsEndpointSimple>
          <AwsEndpointSimple>
            <regionCode>cn-northwest-1</regionCode>
          </AwsEndpointSimple>
        </add>
      </endpoints>
      <isChinaConfigured>true</isChinaConfigured>
      <disabled>false</disabled>
    </AwsAssetDataConnector>
  </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/a
m/aws_asset_data_connector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAssetDataConnector>
      <id>136605</id>
```

```
<name>cn-conn1</name>
<awsAccountId>205767712438</awsAccountId>
<connectorState>QUEUED</connectorState>
<type>AWS</type>
<defaultTags>
<list/>
  </defaultTags>
  <activation>
<list/>
</activation>
<disabled>>false</disabled>
<isGovCloudConfigured>>false</isGovCloudConfigured>
<isChinaConfigured>>true</isChinaConfigured>
  <authRecord>
    <id>132601</id>
    <name>china-auth</name>
  </authRecord>
  <endpoints>
<list>
  <AwsEndpointSimple>
    <regionCode>cn-north-1</regionCode>
  </AwsEndpointSimple>
  <AwsEndpointSimple>
    <regionCode>cn-northwest-1</regionCode>
  </AwsEndpointSimple>
</list>
</endpoints>
<allRegions>>false</allRegions>
</AwsAssetDataConnector>
</data>
</ServiceResponse>
```

Sample 3 : Create new AWS asset data connector for Bahrain (me-south-1)

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-
binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/create/am/awsassetdata
connector/" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>ARN_Global</name>
```

```
<arn>arn:aws:iam::705355653965:role/Demo-PODs</arn>
<externalId></externalId>
<allRegions>true</allRegions>
  <activation>
    <set>
      <ActivationModule>VM</ActivationModule>
      <ActivationModule>CERTVIEW</ActivationModule>
    </set>
  </activation>
</AwsAssetDataConnector>
</data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/a
m/aws_asset_data_connector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <AwsAssetDataConnector>
      <id>566601</id>
      <name>ARN_Global</name>
      <awsAccountId>705355653965</awsAccountId>
      <connectorState>QUEUED</connectorState>
      <type>AWS</type>
      <activation>
        <list>
          <ActivationModule>CERTVIEW</ActivationModule>
          <ActivationModule>VM</ActivationModule>
        </list>
      </activation>
      <disabled>false</disabled>
      <isGovCloudConfigured>false</isGovCloudConfigured>
      <isChinaConfigured>false</isChinaConfigured>
      <arn>arn:aws:iam::705355653965:role/Demo-PODs</arn>
      <externalId>1550261312725</externalId>
      <qualysAwsAccountId>383031258652</qualysAwsAccountId>
      <authRecord/>
      <endpoints>
        <list>
          <AwsEndpointSimple>
            <regionCode>us-west-1</regionCode>
          </AwsEndpointSimple>
          <AwsEndpointSimple>
            <regionCode>ap-northeast-1</regionCode>
          </AwsEndpointSimple>
        </list>
      </endpoints>
    </AwsAssetDataConnector>
  </data>
</ServiceResponse>
```

```
...
    <AwsEndpointSimple>
      <regionCode>me-south-1</regionCode>
    </AwsEndpointSimple>
  </list>
</endpoints>
  <allRegions>true</allRegions>
</AwsAssetDataConnector>
</data>
</ServiceResponse>
```

Sample 4 : Update an AWS asset data connector to include Bahrain (me-south-1)

API request:

```
curl -u "USERNAME:PASSWORD" -H "Content-type: text/xml" -X "POST" --data-
binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/update/am/awsassetdata
connector/12345"
< file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ServiceRequest>
  <data>
    <AwsAssetDataConnector>
      <name>Updated Name</name>
      <defaultTags>
        <add>
          <TagSimple>
            <id>1</id>
          </TagSimple>
        </add>
      </defaultTags>
      <endpoints>
        <add>
          <AwsEndpointSimple>
            <regionCode>us-east-1</regionCode>
          </AwsEndpointSimple>
          <AwsEndpointSimple>
            <regionCode>me-south-1</regionCode>
          </AwsEndpointSimple>
        </add>
      </endpoints>
    </AwsAssetDataConnector>
  </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/a
m/awsassetdataconnector.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <AssetDataConnector>
      <id>12345</id>
      <name>External VPC</name>
      <lastSync>2014-11-26T08:44:05Z</lastSync>
      <lastError />
      <connectorState>SUCCESS</connectorState>
      <type>AWS</type>
      <defaultTags>
        <list>
          <TagSimple>
            <id>1</id>
            <name>EC2</name>
          </TagSimple>
        </list>
      </defaultTags>
      <activation>
        <ActivationModule>VM</ActivationModule>
      </activation>
      <disabled>false</disabled>
      <isGovCloudConfigured>false</isGovCloudConfigured>
    </AssetDataConnector>
  </data>
</ServiceResponse>
```

Host Asset API: Search IBM assets

This feature is coming up in a later release of the Cloud Agent.

SAQ - New Campaign API

API affected	/qps/rest/1.0/count/saq/questionnairecampaign/ [GET][POST] /qps/rest/1.0/search/saq/questionnairecampaign/ [POST] /qps/rest/1.0/get/saq/questionnairecampaign/<id> [GET] /qps/rest/1.0/status/saq/questionnairecampaign/<id> [GET] /qps/rest/1.0/create/saq/questionnairecampaign/ [POST] /qps/rest/1.0/update/saq/questionnairecampaign/<id> [POST] /qps/rest/1.0/delete/saq/questionnairecampaign/<id> [POST] /qps/rest/1.0/launch/saq/questionnairecampaign/<id> [POST] /qps/rest/1.0/complete/saq/questionnairecampaign/<id> [POST] /qps/rest/1.0/cancel/saq/questionnairecampaign/<id> [POST] /qps/rest/1.0/summary/saq/questionnairecampaign/ [POST]
New or Updated APIs	New
DTD or XSD changes	New
API affected	/qps/rest/1.0/search/saq/questionnaire
New or Updated APIs	New
DTD or XSD changes	New

Introducing 2 new APIs for SAQ:

1) Questionnaire Campaign API

<https://<baseurl>/qps/rest/1.0/{action}/saq/questionnairecampaign>

This API will give the list of campaigns to the authorized user. Currently, we support these actions for the campaign: search, count, view status, create, delete, update, launch, complete, cancel, view summary and get details of a campaign.

2) Questionnaire API

<https://<baseurl>/qps/rest/1.0/{action}/saq/questionnaire>

This API will give the list of questionnaires for the campaigns to the authorized user.

The updated Qualys Security Assessment Questionnaire API User Guide will be provided at the time of release. This guide has complete details and samples to help you with using the SAQ API.

Search Campaign

Search for a campaign(s) using specified filters.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have "API ACCESS" permission, Output includes campaigns within the API user's scope. Campaign must be within the API user's scope. Input Parameters

Element	Description
id (Long)	(Optional) Id of the campaign. Supports "EQUALS" and "IN" operators. For example, <Criteria field="id" operator="EQUALS">1234</Criteria>
name (string)	(Optional) name of the campaign. Supports "EQUALS" and "CONTAINS" operators. For example, <Criteria field="name" operator="EQUALS">campaign 1</Criteria>
status (string)	(Optional) Status of the campaign. The values are ACTIVE, INACTIVE, COMPLETE, CANCELED. Supports "EQUALS" and "IN" operators. For example, <Criteria field="status" operator="EQUALS">ACTIVE</Criteria>
workflowType (string)	(Optional) Workflow type of the campaign. The values are VERY_HIGH, HIGH, MEDIUM, LOW. Supports "EQUALS" and "IN" operators. For example, <Criteria field="workflowType" operator="EQUALS">LOW</Criteria>

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/search/saq/questionnairecampaign/"<file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="EQUALS">NCSC Basic Cyber Security
Controls (BCSC) Version 1.0</Criteria>
  </filters>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/sa
q/questionnairecampaign.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <QuestionnaireCampaign>
      <id>565002</id>
      <name>NCSC Basic Cyber Security Controls (BCSC) Version
1.0</name>
      <workflowType>SIMPLE</workflowType>
      <status>CANCELED</status>
      <comments>demo api cancel </comments>
      <created>2018-12-10T04:58:50Z</created>
      <updated>2019-09-25T10:45:26Z</updated>
      <launchDate>2018-12-10T04:58:57Z</launchDate>
      <template>
        <id>624001</id>
        <name>NCSC Basic Cyber Security Controls (BCSC) Version
1.0</name>
        <questionCnt>135</questionCnt>
      </template>
      <user>
        <id>70596886</id>
        <uuid>3d124658-8f4f-771d-806d-1e18a04a1acf</uuid>
        <firstName>api-user</firstName>
        <lastName>api-user</lastName>
        <emailAddress>user@email.com</emailAddress>
        <userName>user</userName>
        <title>dev</title>
        <tags>
          <Tag>
            <id>101271291</id>
            <name>Unassigned Business Unit</name>
          </Tag>
        </tags>
      </user>
      <campaignNotification>
        <reminderNotification>
          <enabled>false</enabled>
          <interval>1</interval>
          <noOfDays>1</noOfDays>
        </reminderNotification>
        <overdueNotification>
          <enabled>false</enabled>
```

```

        <interval>1</interval>
        <noOfDays>1</noOfDays>
    </overdueNotification>
    <dueDateNotification>
        <enabled>false</enabled>
        <interval>1</interval>
    </noOfDays>1</noOfDays>
    </dueDateNotification>
</campaignNotification>
</QuestionnaireCampaign>
</data>
</ServiceResponse>

```

Count Campaign

Show total count of campaigns.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have "API ACCESS", "Access SAQ module" permissions. Campaign must be within the API user's scope.

Input Parameters

Element	Description
id (Long)	(Optional) Id of the campaign. Supports "EQUALS" and "IN" operators. For example, <Criteria field="id" operator="EQUALS">1234</Criteria>
name (string)	(Optional) name of the campaign. Supports "EQUALS" and "CONTAINS" operators. For example, <Criteria field="name" operator="EQUALS">campaign 1</Criteria>
status (string)	(Optional) Status of the campaign. The values are ACTIVE, INACTIVE, COMPLETE, CANCELED. Supports "EQUALS" and "IN" operators. For example, <Criteria field="status" operator="EQUALS">ACTIVE</Criteria>
workflowType (string)	(Optional) Workflow type of the campaign. The values are SIMPLE, REVIEWABLE, FULL. Supports "EQUALS" and "IN" operators. For example, <Criteria field="workflowType" operator="EQUALS">FULL</Criteria>

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/count/saq/questionnairecampaign/"<file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="status" operator="EQUALS">COMPLETE</Criteria>
  </filters>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/saq/questionnairecampaign.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>35</count>
</ServiceResponse>
```

Get Campaign Details

Get the details of a campaign.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have "API ACCESS" permission, Output includes campaigns within the API user's scope. Campaign must be within the API user's scope.

Input Parameters

Element	Description
id (Long)	(Required) Id of the campaign.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "GET"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/get/saq/questionnairecampaign/640002"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/sa
q/questionnairecampaign.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <QuestionnaireCampaign>
      <id>640002</id>
      <name>Test-Campaign-KT</name>
      <workflowType>FULL</workflowType>
      <status>COMPLETE</status>
      <created>2019-02-01T05:47:16Z</created>
      <updated>2019-02-01T05:47:50Z</updated>
      <launchDate>2019-02-01T05:47:50Z</launchDate>
      <template>
        <id>722010</id>
        <name>Test-Template-KT</name>
        <questionCnt>19</questionCnt>
      </template>
      <user>
        <id>70596886</id>
        <uuid>3d124658-8f4f-771d-806d-1e18a04a1acf</uuid>
        <firstName>john</firstName>
        <lastName>doe</lastName>
        <emailAddress>user@email.com</emailAddress>
        <userName>user</userName>
        <title>QA</title>
        <tags>
          <Tag>
            <id>101271291</id>
            <name>Unassigned Business Unit</name>
          </Tag>
        </tags>
      </user>
      <reviewer>
        <id>71077391</id>
        <uuid>831b7266-08f5-4a10-afdd-ab9453075355</uuid>
        <firstName>api-user</firstName>
        <lastName>api-user</lastName>
        <company>afko</company>
        <emailAddress>user@email.com</emailAddress>
        <userName>user</userName>
        <title>reviewer</title>
      </reviewer>
      <approver>
        <id>89375990</id>
```

```

        <uuid>76fcec21-f674-4e17-a868-801f7a92c35c</uuid>
        <firstName>john</firstName>
        <lastName>doe</lastName>
        <company>afko</company>
        <emailAddress>user@email.com</emailAddress>
        <userName>user</userName>
    </approver>
    <campaignNotification>
        <reminderNotification>
            <enabled>>false</enabled>
            <interval>1</interval>
            <noOfDays>1</noOfDays>
        </reminderNotification>
        <overdueNotification>
            <enabled>>false</enabled>
            <interval>1</interval>
            <noOfDays>1</noOfDays>
        </overdueNotification>
        <dueDateNotification>
            <enabled>>false</enabled>
            <interval>1</interval>
            <noOfDays>1</noOfDays>
        </dueDateNotification>
    </campaignNotification>
</QuestionnaireCampaign>
</data>
</ServiceResponse>

```

Get Campaign Status

Get the status of a campaign.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have “API ACCESS” permission, Output includes campaigns within the API user's scope. Campaign must be within the API user's scope.

Input Parameters

Element	Description
id (Long)	(Required) Id of the campaign.

API request:

```

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "GET"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/status/saq/questionnairecampai
gn/640002"

```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/sa
q/questionnairecampaign.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <QuestionnaireCampaign>
      <id>926001</id>
      <name>Demo-campaign-api-1</name>
      <status>ACTIVE</status>
    </QuestionnaireCampaign>
  </data>
</ServiceResponse>
```

Create Campaign

Create a new campaign.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have “API ACCESS”, “Access SAQ module”, and “Questionnaire Campaign Create” permissions.

Input Parameters

Element	Description
name (Text)	(Required) Name of the campaign.
template.id (Long)	(Required) ID of the template that is used for creating the campaign.
due date (Date)	(Required) Due date for the recipients to complete the campaign.
reviewer.id (Long)	(Optional) ID of the campaign reviewer. Required if the workflow type is Reviewable or Full.
approver.id (Long)	(Optional) ID of the approver. Required if the workflow type is Full.
campaignNotification	List of notification types for the campaign (reminder, due date and overdue.

Element	Description
workflowType (Text)	(Required) Workflow type of the campaign. The values are: SIMPLE, REVIEWABLE and FULL. - SIMPLE - When specified the campaign gets completed in two stages. The campaign is sent to the specified recipients and responses are collected from the recipients. - REVIEWABLE - The campaign gets completed in three stages. The first two stages are similar to the Simple workflow. In the third stage the collected responses are sent to a specified reviewer for review. FULL- The campaign gets completed in four stages. The first three stages are similar to Reviewable workflow. In the fourth stage the reviewed responses are sent to a specified approver for approval. After approver's approval, the campaign gets completed.
reminderNotification.enabled (Boolean)	(Optional) True if notification enabled flag is set to true for the campaign.
reminderNotification.noOfDays (Integer)	(Optional) Number of days after which first reminder notification will be sent.
reminderNotification.interval (Integer)	(Optional) The interval in days after which next reminder notification will be sent.
dueDateNotification.enabled (Boolean)	(Optional) True if due date notification enabled flag is set to true for the campaign.
dueDateNotification.noOfDays (Integer)	(Optional) Number of days after which first reminder notification for due date of campaign will be sent.
dueDateNotification.interval (Integer)	(Optional) The interval in days after which the next reminder notification for campaign due date will be sent.
overdueNotification.enabled (Boolean)	(Optional) True if overdue notification enabled flag is set to true for the campaign.
overdueNotification.noOfDays (Integer)	(Optional) Number of days after which first reminder notification for campaign getting overdue will be sent.
overdueNotification.interval (Integer)	(Optional) The interval in days after which the next reminder notification for campaign getting overdue will be sent.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/create/saq/questionnairecampai
gn/" <file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <data>
    <QuestionnaireCampaign>
      <name>Test API demo 1</name>
      <template>
        <id>122601</id>
      </template>
      <workflowType>REVIEWABLE</workflowType>
      <dueDate>2019-09-28T17:08:00Z</dueDate>
      <reviewer>
        <id>352088</id>
      </reviewer>
      <campaignNotification>
        <reminderNotification>
          <enabled>true</enabled>
          <interval>2</interval>
          <noOfDays>2</noOfDays>
        </reminderNotification>
        <overdueNotification>
          <enabled>false</enabled>
        </overdueNotification>
      </campaignNotification>
    </QuestionnaireCampaign>
  </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/sa
q/questionnairecampaign.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <QuestionnaireCampaign>
      <id>120802</id>
```

```
<name>Test API demo 1</name>
<workflowType>REVIEWABLE</workflowType>
<status>INACTIVE</status>
<dueDate>2019-09-28T17:08:00Z</dueDate>
<created>2019-09-25T10:24:16Z</created>
<updated>2019-09-25T10:24:16Z</updated>
<template>
  <id>122601</id>
</template>
<user>
  <id>337590</id>
</user>
<reviewer>
  <id>352088</id>
</reviewer>
<campaignNotification>
  <reminderNotification>
    <enabled>true</enabled>
    <interval>2</interval>
    <noOfDays>2</noOfDays>
  </reminderNotification>
  <overdueNotification>
    <enabled>false</enabled>
    <interval>1</interval>
    <noOfDays>1</noOfDays>
  </overdueNotification>
  <dueDateNotification>
    <enabled>false</enabled>
    <interval>1</interval>
    <noOfDays>1</noOfDays>
  </dueDateNotification>
</campaignNotification>
</QuestionnaireCampaign>
</data>
</ServiceResponse>
```

Update Campaign

Update a campaign.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have "API ACCESS" permission, Output includes users within the API user's scope.

Input Parameters

Element	Description
id (Long)	(Required) Name of the campaign.
name (Text)	(Optional) Name of the campaign.
template.id (Long)	(Optional) ID of the template that is used for creating the campaign.
workflowType (String)	(Optional) Workflow type of the campaign. The values are: SIMPLE, REVIEWABLE and FULL. - SIMPLE - When specified the campaign gets completed in two stages. The campaign is sent to the specified recipients and responses are collected from the recipients. - REVIEWABLE - The campaign gets completed in three stages. The first two stages are similar to the Simple workflow. In the third stage the collected responses are sent to a specified reviewer for review. FULL- The campaign gets completed in four stages. The first three stages are similar to Reviewable workflow. In the fourth stage the reviewed responses are sent to a specified approver for approval. After approver's approval, the campaign gets completed.
due date (Date)	(Optional) Due date for the recipients to complete the campaign.
reviewer.id (Long)	(Optional) ID of the campaign reviewer. Required if the workflow type is Reviewable or Full.
approver.id (Long)	(Optional) ID of the approver. Required if the workflow type is Full.
campaignNotification	List of notification types for the campaign (reminder, due date and overdue).
reminderNotification.enabled (Boolean)	(Optional) True if notification enabled flag is set to true for the campaign.
reminderNotification.noOfDays (Integer)	(Optional) Number of days after which first reminder notification will be sent.

Element	Description
reminderNotification.interval (Integer)	(Optional) The interval in days after which next reminder notification will be sent.
dueDateNotification.enabled (Boolean)	(Optional) True if due date notification enabled flag is set to true for the campaign.
dueDateNotification.noOfDays (Integer)	(Optional) Number of days after which first reminder notification for due date of campaign will be sent.
dueDateNotification.interval (Integer)	(Optional) The interval in days after which the next reminder notification for campaign due date will be sent.
overdueNotification.enabled (Boolean)	(Optional) True if overdue notification enabled flag is set to true for the campaign.
overdueNotification.noOfDays (Integer)	(Optional) Number of days after which first reminder notification for campaign getting overdue will be sent.
overdueNotification.interval (Integer)	(Optional) The interval in days after which the next reminder notification for campaign getting overdue will be sent.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/update/saq/questionnairecampai
gn/" <file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <data>
    <QuestionnaireCampaign>
      <id>120802</id>
      <workflowType>FULL</workflowType>
      <approver>
        <id>352088</id>
      </approver>
      <reviewer>
        <id>352088</id>
      </reviewer>
      <recipients>
        <add>
          <User>
            <id>339590</id>
            <assigned>New </assigned>
          </User>
        </add>
      </recipients>
    </QuestionnaireCampaign>
  </data>
</ServiceRequest>
```

```

        <User>
            <id>339591</id>
            <assigned>New </assigned>
        </User>
    </add>
</recipients>
<campaignNotification>
    <overdueNotification>
        <enabled>true</enabled>
        <interval>1</interval>
        <noOfDays>5</noOfDays>
    </overdueNotification>
    <dueDateNotification>
        <enabled>false</enabled>
    </dueDateNotification>
</campaignNotification>
</QuestionnaireCampaign>
</data>
</ServiceRequest>

```

XML output:

```

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/sa
q/questionnairecampaign.xsd">
<responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <QuestionnaireCampaign>
            <id>120802</id>
            <name>Test API demo 1</name>
            <workflowType>FULL</workflowType>
            <status>INACTIVE</status>
            <dueDate>2019-09-28T17:08:00Z</dueDate>
            <created>2019-09-25T10:24:16Z</created>
            <updated>2019-09-25T10:26:56Z</updated>
            <template>
                <id>122601</id>
            </template>
            <user>
                <id>337590</id>
            </user>
            <reviewer>
                <id>352088</id>
            </reviewer>
            <approver>
                <id>352088</id>
            </approver>
        </QuestionnaireCampaign>
    </data>
</ServiceResponse>

```

```
<campaignNotification>
  <reminderNotification>
    <enabled>true</enabled>
    <interval>2</interval>
    <noOfDays>2</noOfDays>
  </reminderNotification>
  <overdueNotification>
    <enabled>true</enabled>
    <interval>1</interval>
    <noOfDays>5</noOfDays>
  </overdueNotification>
  <dueDateNotification>
    <enabled>false</enabled>
    <interval>1</interval>
    <noOfDays>1</noOfDays>
  </dueDateNotification>
</campaignNotification>
</QuestionnaireCampaign>
</data>
</ServiceResponse>
```

Delete Campaign

Delete a campaign.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have “API ACCESS”, “Access SAQ module”, and “Questionnaire Campaign Delete” permissions. Campaign must be within the API user's scope.

Input Parameters.

Element	Description
id (Long)	(Required) Id of the campaign.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/delete/saq/questionnairecampaign/952401"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/saq/questionnairecampaign.xsd">
  <responseCode>SUCCESS</responseCode>
```

```
<count>1</count>
<data>
  <QuestionnaireCampaign>
    <id>952401</id>
  </QuestionnaireCampaign>
</data>
</ServiceResponse>
```

Launch Campaign

Launch a new campaign. You can launch only campaign that are inactive.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have "API ACCESS", "Access SAQ module", and "Questionnaire Campaign Launch" permissions. Campaign must be within the API user's scope.

Input Parameters.

Element	Description
id (Long)	(Required) Id of the campaign.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/launch/saq/questionnairecampai
gn/640002"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/sa
q/questionnairecampaign.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <QuestionnaireCampaign>
      <id>640002</id>
      <name>Test-Campaign-KT</name>
      <workflowType>FULL</workflowType>
      <status>COMPLETE</status>
      <created>2019-02-01T05:47:16Z</created>
      <updated>2019-02-01T05:47:50Z</updated>
      <launchDate>2019-02-01T05:47:50Z</launchDate>
      <template>
        <id>722010</id>
        <name>Test-Template-KT</name>
```

```

        <questionCnt>19</questionCnt>
    </template>
    <user>
        <id>70596886</id>
        <uuid>3d124658-8f4f-771d-806d-1e18a04a1acf</uuid>
        <firstName>api-user</firstName>
        <lastName>api-user</lastName>
        <emailAddress>user@email.com</emailAddress>
        <userName>user</userName>
        <title>dev</title>
        <tags>
            <Tag>
                <id>101271291</id>
                <name>Unassigned Business Unit</name>
            </Tag>
        </tags>
    </user>
    <reviewer>
        <id>71077391</id>
        <uuid>831b7266-08f5-4a10-afdd-ab9453075355</uuid>
        <firstName>api-user</firstName>
        <lastName>api-user</lastName>
        <company>afko</company>
        <emailAddress>user@email.com</emailAddress>
        <userName>user</userName>
        <title>reviewer</title>
    </reviewer>
    <approver>
        <id>89375990</id>
        <uuid>76fcec21-f674-4e17-a868-801f7a92c35c</uuid>
        <firstName>john</firstName>
        <lastName>doe</lastName>
        <company>afko</company>
        <emailAddress>user@email.com</emailAddress>
        <userName>user</userName>
    </approver>
    <campaignNotification>
        <reminderNotification>
            <enabled>false</enabled>
            <interval>1</interval>
            <noOfDays>1</noOfDays>
        </reminderNotification>
        <overdueNotification>
            <enabled>false</enabled>
            <interval>1</interval>
            <noOfDays>1</noOfDays>
        </overdueNotification>
        <dueDateNotification>
            <enabled>false</enabled>

```

```

        <interval>1</interval>
        <noOfDays>1</noOfDays>
    </dueDateNotification>
</campaignNotification>
</QuestionnaireCampaign>
</data>
</ServiceResponse>

```

Complete Campaign

Mark a campaign complete. Only active campaigns can be marked complete.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have "API ACCESS", "Access SAQ module", and "Questionnaire Campaign Complete" permissions. Campaign must be within the API user's scope.

Input Parameters

Element	Description
id (Long)	(Required) Id of the campaign.

API request:

```

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/complete/saq/questionnairecamp
aign/684001"

```

XML output:

```

<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="qualysapi.qualys.com/qps/xsd/1.0/saq/quest
ionnairecampaign.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <QuestionnaireCampaign>
            <id>684001</id>
            <name>Test10_vinod</name>
            <status>COMPLETE</status>
        </QuestionnaireCampaign>
    </data>
</ServiceResponse>

```

Cancel Campaign

Mark a campaign cancelled. Only active campaigns can be marked cancelled.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have "API ACCESS", "Access SAQ module", and "Questionnaire Campaign Cancel" permissions. Campaign must be within the API user's scope.

Input Parameters

Element	Description
id (Long)	(Required) Id of the campaign.
comments (Text)	(Required) Comments for cancelling the campaign.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/cancel/saq/questionnairecampai
gn/565002"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/sa
q/questionnairecampaign.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <QuestionnaireCampaign>
      <id>565002</id>
      <name>NCSC Basic Cyber Security Controls (BCSC) Version
1.0</name>
      <status>CANCELED</status>
    </QuestionnaireCampaign>
  </data>
</ServiceResponse>
```

View Campaign Summary

View the details of a campaign.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have "API ACCESS", "Access SAQ module", and "Questionnaire Campaign View" permissions. Campaign must be within the API user's scope.

Input Parameters

Element	Description
id (Long)	(Required) Id of the campaign.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/summary/saq/questionnairecampaign/684001"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/saq/questionnairecampaign.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <QuestionnaireCampaign>
      <name>Campaign_RevBP</name>
      <workflowType>REVIEWABLE</workflowType>
      <template>
        <name>ASQTe</name>
      </template>
      <reviewer>
        <id>71077391</id>
        <userName>reviewer.r.39b9xw@qualys</userName>
      </reviewer>
      <summary>
        <count>2</count>
        <list>
          <QuestionnaireSummary>
            <state>Closed</state>
            <value>1</value>
          </QuestionnaireSummary>
          <QuestionnaireSummary>
            <state>In Progress</state>
```

```

        <value>1</value>
      </QuestionnaireSummary>
    </list>
  </summary>
</QuestionnaireCampaign>
</data>
</ServiceResponse>

```

Search Questionnaires

Search questionnaires of a campaign.

Permissions - User must have the Security Assessment Questionnaire (SAQ) module enabled, User must have "API ACCESS", "Access SAQ module", and "Questionnaire Campaign" permissions. Campaign must be within the API user's scope.

Input Parameters

Element	Description
id (Long)	(Optional) Id of the questionnaire. Supports "EQUALS" and "IN" operators. For example, <Criteria field="id" operator="EQUALS">1234</Criteria>
campaign.id (Long)	(Optional) Id of the campaign. Supports "EQUALS" and "IN" operators. For example, <Criteria field="campaign.id" operator="EQUALS">4527</Criteria>
state (string)	(Optional) State of the questionnaire. The values are INFORMATION_GATHERING, APPROVAL, REVIEW, CANCELED, INCOMPLETE, CLOSED. Supports "EQUALS" and "IN" operators. For example, <Criteria field="state" operator="EQUALS">REVIEW</Criteria>
risk (string)	(Optional) Risk value assigned to the questionnaire. The values are VERY_HIGH, HIGH, MEDIUM, LOW. Supports "EQUALS" and "IN" operators. For example, <Criteria field="risk" operator="LOW">REVIEW</Criteria>

API request:

```

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/1.0/search/saq/questionnaire/"

```

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="state" operator="EQUALS">APPROVAL</Criteria>
  </filters>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/1.0/sa
q/questionnaire.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <hasMoreRecords>>false</hasMoreRecords>
  <data>
    <Questionnaire>
      <id>707802</id>
      <title>Test-campaign-demo1 - responder r</title>
      <dueDate>2019-02-22T00:00:00Z</dueDate>
      <assignedDate>2019-02-12T06:31:48Z</assignedDate>
      <createdBy>
        <id>70596886</id>
        <uuid>3d124658-8f4f-771d-806d-1e18a04a1acf</uuid>
        <firstName>api-user</firstName>
        <lastName>api-user</lastName>
        <company>afko</company>
        <emailAddress>user@email.com</emailAddress>
        <userName>user</userName>
        <title>QA</title>
        <tags>
          <Tag>
            <id>101271291</id>
            <name>Unassigned Business Unit</name>
          </Tag>
        </tags>
      </createdBy>
      <updatedBy>
        <id>89375990</id>
        <uuid>76fcec21-f674-4e17-a868-801f7a92c35c</uuid>
        <firstName>api-user</firstName>
        <lastName>api-user</lastName>
        <company>afko</company>
        <emailAddress>user@email.com</emailAddress>
        <userName>user</userName>
        <title>dev</title>
```

```

</updatedAt>
<assignedTo>
  <id>89375990</id>
  <uuid>76fcec21-f674-4e17-a868-801f7a92c35c</uuid>
  <firstName>john</firstName>
  <lastName>doe</lastName>
  <company>afko</company>
  <emailAddress>user@email.com</emailAddress>
  <userName>user</userName>
</assignedTo>
<responder>
  <id>71077390</id>
  <uuid>97318a7b-85a8-466d-8071-597cde68e1b4</uuid>
  <firstName>resonder1</firstName>
  <lastName>doe</lastName>
  <company>afko</company>
  <emailAddress>user@email.com</emailAddress>
  <userName>user</userName>
  <title>responder</title>
</responder>
<originalAssignedTo>
  <id>71077390</id>
  <uuid>97318a7b-85a8-466d-8071-597cde68e1b4</uuid>
  <firstName>responder1</firstName>
  <lastName>doe</lastName>
  <company>afko</company>
  <emailAddress>user@email.com</emailAddress>
  <userName>user</userName>
  <title>tag1</title>
</originalAssignedTo>
<campaign>
  <id>656002</id>
</campaign>
<maxPossibleScore>0</maxPossibleScore>
<score>0</score>
<state>APPROVAL</state>
<progress>
  <total>14</total>
  <unanswered>0</unanswered>
  <answered>14</answered>
  <delegated>0</delegated>
  <qaacceptedQuestions>14</qaacceptedQuestions>
</progress>
</Questionnaire>
</data>
</ServiceResponse>

```

WAS: Tag Details in Web App API

API affected	/qps/rest/3.0/search/was/webapp [POST]
New or Updated APIs	Updated API
DTD or XSD changes	Yes

With introduction of new optional parameter for Web Applications API, you can now also view the list of tags (and not just count of tags) associated with the web application.

Permissions

- The user must have the WAS module enabled.
- The "API Access" and "Access WAS module" permissions must be enabled for the user
- The web application must be within the user's scope.

Input Parameters

Parameter	Description
verbose (boolean)	A flag to indicate whether the list of tags associated with the web application should be listed or not. Example: <pre><preferences> <verbose>true</verbose> </preferences></pre>

Sample - Search Web Application and view associated tags

Let us search for a web application and view tags associated with the web application.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @- "https://qualysapi.qualys.com/qps/rest/3.0/search/was/webapp" <file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data (file.xml):

```
<ServiceRequest>
  <preferences>
    <verbose>true</verbose>
  </preferences>
  <filters>
    <Criteria field="name" operator="CONTAINS">My Web
```

```
Application</Criteria>
  </filters>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w
as/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WebApp>
      <id>6620298</id>
      <name>
        <![CDATA[My Web Application]]>
      </name>
      <url>
        <![CDATA[http://www.example.com]]>
      </url>
      <owner>
        <id>1056860</id>
      </owner>
      <tags>
        <count>1</count>
        <list>
          <Tag>
            <id>9029017</id>
            <name>
              <![CDATA[TagWebapp1]]>
            </name>
          </Tag>
        </list>
      </tags>
      <createdDate>2017-12-15T16:13:06Z</createdDate>
      <updatedAtDate>2018-11-19T04:38:08Z</updatedAtDate>
    </WebApp>
  </data>
</ServiceResponse>
```

XSD Update

Changes in webapp.xsd (.../qps/xsd/3.0/was/webapp.xsd).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
```

```
...  
<xs:complexType name="ServiceRequestPreferences">  
  <xs:all>  
    <xs:element name="startFromId" type="xs:long" minOccurs="0"/>  
    <xs:element name="startFromOffset" type="xs:int" minOccurs="0"/>  
    <xs:element name="limitResults" type="xs:int" minOccurs="0"/>  
    <xs:element name="verbose" type="xs:boolean" minOccurs="0"/>  
  </xs:all>  
</xs:complexType>  
...  
  </xs:all>  
</xs:complexType>  
</xs:schema>
```

WAS: XSS Payloads Option for Standard Scans

API affected	/qps/rest/3.0/create/was/optionprofile [POST] /qps/rest/3.0/update/was/optionprofile/<id> [POST]
New or Updated APIs	Updated API
DTD or XSD changes	Yes

You can now enable comprehensive tests for cross-site scripting vulnerabilities to be executed during our standard scan using the new parameter in option profile. The comprehensive tests includes XSS with exhaustive set of payloads including set of standard payloads. Running a scan with XSS payloads option enabled in the detection scope of standard scan will provide the best assurance that your web application is free from XSS vulnerabilities. However, enabling this option leads to significant increase in the scan time.

Permissions

- The user must have the WAS module enabled.
- The "API Access" and "Create Option Profile" permissions must be enabled for the user
- The web application must be within the user's scope.

Input Parameters

Parameter	Description
enableXssPayloads (boolean)	A flag to indicate if XSS payloads should be enabled or disabled during the scan. If the flag is set to true, comprehensive tests for cross-site scripting vulnerabilities are executed during the scan Example: <pre><detection> <detectionScope>CORE</detectionScope> <enableXssPayloads>true</enableXssPayloads> </detection></pre>

Sample - Create an option profile with XSS payloads enabled

Let us create an option profile to launch a standard scan with comprehensive tests for cross-site scripting vulnerabilities enabled.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --  
databinary@-  
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/optionprofile" <  
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data (file.xml):

```
<ServiceRequest>  
  <data>  
    <OptionProfile>  
      <name>Sample Option Profile With XSS Payloads</name>  
      <detection>  
        <detectionScope>CORE</detectionScope>  
        <enableXssPayloads>true</enableXssPayloads>  
      </detection>  
    </OptionProfile>  
  </data>  
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w  
as/optionprofile.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <OptionProfile>  
      <id>81333</id>  
      <name>  
        <![CDATA[Launch Scan with XSS Payloads enabled]]>  
      </name>  
      <owner>  
        <id>412791</id>  
        <username>user_john</username>  
        <firstName><![CDATA[John]]></firstName>  
        <lastName><![CDATA[Doe]]></lastName>  
      </owner>  
      <isDefault>>false</isDefault>  
    </OptionProfile>  
  </data>  
</ServiceResponse>
```

```
<count>0</count>
</tags>
<formSubmission>BOTH</formSubmission>
<maxCrawlRequests>300</maxCrawlRequests>
<timeoutErrorThreshold>100</timeoutErrorThreshold>
<unexpectedErrorThreshold>300</unexpectedErrorThreshold>
<parameterSet>
  <id>0</id>
  <name>
    <![CDATA[Initial Parameters]]>
  </name>
</parameterSet>
<ignoreBinaryFiles>>false</ignoreBinaryFiles>
<includeActionUriInFormId>>false</includeActionUriInFormId>
<enhancedCrawling>>false</enhancedCrawling>
<smartScanSupport>>false</smartScanSupport>
<performance>LOW</performance>
<bruteforceOption>MINIMAL</bruteforceOption>
<detection>
  <b><detecionScope>CORE</detecionScope>
  <b><enableXssPayloads>true</enableXssPayloads>
</detection>
<comments>
  <count>0</count>
</comments>
<sensitiveContent>
  <creditCardNumber>>false</creditCardNumber>
  <socialSecurityNumber>>false</socialSecurityNumber>
</sensitiveContent>
<createdDate>2019-10-04T11:11:59Z</createdDate>
<createdBy>
<owner>
  <id>412791</id>
  <username>user_john</username>
  <firstName><![CDATA[John]]></firstName>
  <lastName><![CDATA[Doe]]></lastName>
</owner>
</createdBy>
<updatedAt>2018-07-25T03:45:12Z</updatedAt>
<updatedBy>
  <owner>
    <id>412791</id>
    <username>user_john</username>
    <firstName><![CDATA[John]]></firstName>
    <lastName><![CDATA[Doe]]></lastName>
  </owner>
</updatedBy>
</OptionProfile>
</data>
```

```
</ServiceResponse>
```

Sample - Update an existing option profile to enable XSS payloads option

Let us update an existing option profile to enable the XSS payloads option.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --  
databinary@-  
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/16003  
" < file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data (file.xml):

```
<ServiceRequest>  
  <data>  
    <OptionProfile>  
      <name>Sample Option Profile With XSS Payloads</name>  
      <detection>  
        <detectionScope>CORE</detectionScope>  
        <enableXssPayloads>true</enableXssPayloads>  
      </detection>  
    </OptionProfile>  
  </data>  
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>  
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w  
as/optionprofile.xsd"  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <OptionProfile>  
      <id>832265669</id>  
    </OptionProfile>  
  </data>  
</ServiceResponse>
```

XSD Update

Changes in optionprofile.xsd (../qps/xsd/3.0/was/optionprofile.xsd).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  ...
  <xs:element name="enableXssPayloads" type="xs:boolean" minOccurs="0"/>
  ...
  </xs:all>
</xs:complexType>
</xs:schema>
```

WAS: New Groups for Information Gathered Issues

API affected	/qps/rest/3.0/get/was/finding/<id> [GET]
New or Updated APIs	Updated API
DTD or XSD changes	Yes

Currently, all Information Gathered issues in WAS are clubbed together in the report. We have now introduced two new groups for issues of type Information Gathered:

- Diagnostic IG (general information about the scan)
- Weakness IG (issues that are security weakness or conflict with best practices)

The response accordingly reflects to which group the issue belongs.

Permissions

- The user must have the WAS module enabled.
- The "API Access" permission must be enabled for the user
- The web application must be within the user's scope.

Sample - View details for the finding with IG groups

API request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/finding/713223"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w
as/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Finding>
      <id>713223</id>
      <uniqueId>8c9c933f-04f1-f77e-e053-294f2c0ab892</uniqueId>
      <qid>150014</qid>
      <name>
        <![CDATA[External Form Actions Discovered]]>
      </name>
      <type>INFORMATION_GATHERED</type>
      <findingType>QUALYS</findingType>
      <group>IG_DIAG</group>
```

```
        <resultList>
          <count>1</count>
          <list>
            ....
          </tags>
        </webApp>
      </Finding>
    </data>
  </ServiceResponse>
```

XSD Update

Changes in finding.xsd (.../qps/xsd/3.0/was/finding.xsd).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  ...
    <xs:element name="group" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="XSS"/>
          <xs:enumeration value="SQL"/>
          <xs:enumeration value="INFO"/>
          <xs:enumeration value="PATH"/>
          <xs:enumeration value="CC"/>
          <xs:enumeration value="SSN-US"/>
          <xs:enumeration value="CUSTOM"/>
          <xs:enumeration value="IG-DIAG"/>
          <xs:enumeration value="IG-WEAK"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  ...
</xs:schema>
```

WAS: Cancel Scan with Results Support for Scans

API affected	/qps/rest/3.0/cancel/was/wasscan/<id> [POST]
New or Updated APIs	Updated API
DTD or XSD changes	Yes

Currently, canceling an unfinished scan on a web application which is in the user's scope does not return any results. We have now introduced a new parameter `<cancelWithResults>` that allows you to cancel the scan and still retain results. You can use the scan ID and generate a report to view the results.

Permissions

- The user must have the WAS module enabled.
- The "API Access" and "Access WAS module" permission must be enabled for the user
- The web application must be within the user's scope.

Input Parameters

Parameter	Description
<code><cancelWithResults></code> (boolean)	A flag to indicate if the scan to be canceled should retain partial scan results or not. The parameter is supported for single scan, only child scan (but not parent scan). We recommend you to use this parameter only after 20 minutes of scan goes into Running status. Example: <pre><WasScan> <cancelWithResults>true</cancelWithResults> </WasScan></pre>

Sample - Cancel Scan with CancelWithResults option

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/cancel/was/wasscan/6620298"
```

Request POST Data (file.xml):

```
<ServiceRequest>
  <data>
    <WasScan>
      <cancelWithResults>true</cancelWithResults>
    </WasScan>
  </data>
</ServiceRequest>
```

```
    </WasScan>
  </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w
as/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <data>
    <WasScan>
      <id>6620298</id>
    </WasScan>
  </data>
</ServiceResponse>
```

XSD Update

Changes in wasscan.xsd (.../qps/xsd/3.0/was/wasscan.xsd).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  ...
  <xs:element name="sendOneMail" type="xs:boolean" default="false"
minOccurs="0"/>
  <xs:element name="enableWAFAuth" type="xs:boolean" default="false"
minOccurs="0"/>
  <xs:element name="cancelWithResults" type="xs:boolean" default="false"
minOccurs="0"/>
  ...
</xs:schema>
```

WAS: Scan Again Support for Scan API

API affected	/qps/rest/3.0/scanagain/was/scan/<id> [POST]
New or Updated APIs	Updated API
DTD or XSD changes	No

We now provide the option to execute a previous scan again. Identify the scan you want to run again and use `scanagain` action. We'll do our best to pre-fill the scan settings to match the original scan.

Permissions

- The user must have the WAS module enabled.
- The "API Access" and "Access WAS module" permission must be enabled for the user
- The web application must be within the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies the scan to be executed again. You could optionally provide a new name for the scan as well.

Sample - Scan with Scanagain option

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/scanagain/was/wasscan/4626354"
```

Request POST Data (file.xml):

```
<ServiceRequest>
  <data>
    <WasScan>
      <name>Sample Scan Name for Rescan</name>
    </WasScan>
  </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1 </count>
```

```
<data>
  <WasScan>
    <id>4626354</id>
  </WasScan>
</data>
</ServiceResponse>
```