Qualys Cloud Platform (VM, PC) v10.x

API Release Notes

Version 10.5

October 15, 2020

This new version of the Qualys Cloud Platform (VM, PC) includes improvements to the Qualys API. 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

Update Host Attributes for IPs in Custom Networks

New Database UDC for PostgreSQL and Pivotal Greenplum

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.

Update Host Attributes for IPs in Custom Networks

APIs affected	/api/2.0/fo/asset/ip/?action=update
New or Updated API	Updated
DTD or XSD changes	No

Applicable when the Network Support feature is enabled for the subscription.

Previously, if you updated host attributes for an IP address using the API, the IP was updated in the Global Default Network only. Now, with this release, you can update host attributes for an IP address in a custom network. Simply specify the network ID during the update request.

Good to Know

- 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.

- As in previous releases, you can update multiple IPs/ranges in the same request. The host attribute changes will apply to all IPs included in the action.

- Host attributes you can update are: 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.

- 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

We added the input parameter “network_id” in this release. The other input parameters listed below were already supported and have not changed.

Parameter	Description
action=update	(Required)
echo_request={**0**|1}	(Optional) Specify 1 to view (echo) input parameters in the XML output. By default these are not included.

Parameter	Description
ips={value} -or- {POSTed CSV raw data}	(Required) 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}	(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.
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.

API Samples

Example 1 - Update IPs in custom network to change tracking method to DNS

In this sample, network ID 22222 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=22222&ips=10.10.10.200,10.10.23.40&tracking_met
hod=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>2020-10-14T17:27:36Z</DATETIME>
    <TEXT>IPs successfully updated</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Example 2 - Update multiple host attributes in Global Default Network

In this sample, we're updating ud1, ud2, owner, and comments for specified IPs in the Global Default Network (network_id=0).

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=update&network_id=0&ips=10.10.10.17-
10.10.10.25&ud1=USWEST&ud2=IT&owner=qualys_joe&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>
    <TEXT>IPs successfully updated</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Example 3 - Network ID is not in user's scope

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>
```

New Database UDC for PostgreSQL and Pivotal Greenplum

APIs affected	/api/2.0/fo/compliance/posture/info/?action=list /api/2.0/fo/compliance/control/?action=list /api/2.0/fo/compliance/policy/?action=export /api/2.0/fo/subscription/option_profile/pc
New or Updated API	Updated
DTD or XSD changes	Yes

We've introduced a new Database UDC for PostgreSQL/Pivotal Greenplum. For this new database control type, we added new settings in the compliance option profile. You'll see API changes for create, update, list, and export option profiles. We've also added new elements to the XML output and DTDs for Control List Output, Policy Export Output, Posture Info List Output, Option Profiles, and the ImportableControl.xsd schema.

You'll see these changes:

- We've added new input parameters postgreSQL_db_udc_restriction and postgreSQL_db_udc_limit to the Option Profile API to help you set a limit on the number of rows returned per scan for a PostgreSQL/Pivotal Greenplum UDC. The default value is 256 and maximum allowed limit is 5000 rows.

- We've added a new CHECK_TYPE element to the XML output for Control List API: PostgreSQL Database Check.

- We've added support for PostgreSQL technologies (PostgreSQL 9.x, PostgreSQL 10.x, PostgreSQL 11.x, and PostgreSQL 12.x) and Pivotal Greenplum technologies (Pivotal Greenplum 5.x and Pivotal Greenplum 6.x) for the UDC, and you'll see these technologies in Posture API and Policy Export API.

- We've updated the ImportableControl.xsd schema to include a new enumeration value for the CHECK_TYPE element: PostgreSQL Database Check.

Sample - Option Profile API: Create

In this sample, you'll create an option profile and specify the new parameters for PostgreSQL/Pivotal Greenplum: postgreSQL_db_udc_restriction and postgreSQL_db_udc_limit.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=create&title=PostgreOP1&scan_ports=targeted&postgreSQL_db_udc_res
triction=1&postgreSQL_db_udc_limit=50"
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pc/"
```

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>2020-10-15T09:05:39Z</DATETIME>
    <TEXT>Compliance Option profile successfully added.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>4214348</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Option Profile API: Update

In this sample, you'll update an option profile and specify the new parameters for PostgreSQL/Pivotal Greenplum: postgreSQL_db_udc_restriction and postgreSQL_db_udc_limit.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=update&id=4214348&title=PostgreOP1&scan_ports=targeted&postgreSQL
_db_udc_restriction=1&postgreSQL_db_udc_limit=400"
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pc/"
```

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>2020-10-15T10:42:14Z</DATETIME>
    <TEXT>Compliance Option profile successfully updated.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>4214348</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Option Profile API: List

When you list option profiles, you'll see the database preference keys and their corresponding values for PostgreSQL/Pivotal Greenplum.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=list&id=4214350"
"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/opti
on_profile_info.dtd">
<OPTION_PROFILES>
  <OPTION_PROFILE>
    <BASIC_INFO>
      <ID>4214350</ID>
      ...
        <HOSTS_TO_SCAN>
          <EXTERNAL_SCANNERS>15</EXTERNAL_SCANNERS>
          <SCANNER_APPLIANCES>30</SCANNER_APPLIANCES>
        </HOSTS_TO_SCAN>

     ...
      <DATABASE_PREFERENCE_KEY>
         <POSTGRESQL>
           <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
           <DB_UDC_LIMIT>2500</DB_UDC_LIMIT>
        </POSTGRESQL>
      </DATABASE_PREFERENCE_KEY>
      <FILE_INTEGRITY_MONITORING>
        <AUTO_UPDATE_EXPECTED_VALUE>0</AUTO_UPDATE_EXPECTED_VALUE>
      </FILE_INTEGRITY_MONITORING>
    </SCAN>
    ...
    </ADDITIONAL>
  </OPTION_PROFILE>
</OPTION_PROFILES>
```

Sample - Options Profile API: Export

When you export an option profile, you'll see the database preference keys and their corresponding values for PostgreSQL/Pivotal Greenplum.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/?act
ion=export&output_format=xml&option_profile_type=compliance&option_profil
e_id=4214348"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE OPTION_PROFILES SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/opti
on_profile_info.dtd">
<OPTION_PROFILES>
  <OPTION_PROFILE>
...
      </DATABASE_PREFERENCE_KEY>
        <POSTGRESQL>
          <DB_UDC_RESTRICTION>1</DB_UDC_RESTRICTION>
          <DB_UDC_LIMIT>2500</DB_UDC_LIMIT>
        </POSTGRESQL>
      </DATABASE_PREFERENCE_KEY>
      <FILE_INTEGRITY_MONITORING>
        <AUTO_UPDATE_EXPECTED_VALUE>0</AUTO_UPDATE_EXPECTED_VALUE>
      </FILE_INTEGRITY_MONITORING>
    ...
</OPTION_PROFILES>
```

DTD update:

We updated the option_profile_info.dtd to include PostgreSQL/Pivotal Greenplum in Database Preference Key and corresponding elements.

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

```
<!ELEMENT OPTION_PROFILES (OPTION_PROFILE)*>
 <!ELEMENT OPTION_PROFILE (BASIC_INFO, SCAN, MAP?, ADDITIONAL)>
...
<!ELEMENT SCAN (PORTS?, SCAN_DEAD_HOSTS?, CLOSE_VULNERABILITIES?,
PURGE_OLD_HOST_OS_CHANGED?, PERFORMANCE?, LOAD_BALANCER_DETECTION?,
PASSWORD_BRUTE_FORCING?, VULNERABILITY_DETECTION?, AUTHENTICATION?,
        ADDL_CERT_DETECTION?, DISSOLVABLE_AGENT?, LITE_OS_SCAN?,
ETHERNET_IP_PROBING?, CUSTOM_HTTP_HEADER?, HOST_ALIVE_TESTING?,
SCAN_RESTRICTION?, DATABASE_PREFERENCE_KEY?, SYSTEM_AUTH_RECORD?,
FILE_INTEGRITY_MONITORING?, CONTROL_TYPES?, DO_NOT_OVERWRITE_OS?,
```

```
TEST_AUTHENTICATION?)>
 ...

<!ELEMENT DATABASE_PREFERENCE_KEY (MSSQL?, ORACLE?, SYBASE?,
POSTGRESQL?)>
<!ELEMENT MSSQL (DB_UDC_RESTRICTION, DB_UDC_LIMIT)>
<!ELEMENT ORACLE (DB_UDC_RESTRICTION, DB_UDC_LIMIT)>
<!ELEMENT SYBASE (DB_UDC_RESTRICTION, DB_UDC_LIMIT)>
<!ELEMENT POSTGRESQL (DB_UDC_RESTRICTION, DB_UDC_LIMIT)>
<!ELEMENT DB_UDC_RESTRICTION (#PCDATA)>
<!ELEMENT DB_UDC_LIMIT (#PCDATA)>
 ...
```

Schema update (option_profiles.xsd):

The option_profiles.xsd schema is used when importing and exporting option profiles. We added new elements for the PostgreSQL/Pivotal Greenplum database control type.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="OPTION_PROFILES" type="OPTION_PROFILESType"/>
...
    <xs:complexType name="DATABASE_PREFERENCE_KEYType">
        <xs:sequence>
            <xs:element type="MSSQLType" name="MSSQL" minOccurs="0"/>
            <xs:element type="ORACLEType" name="ORACLE" minOccurs="0"/>
            <xs:element type="SYBASEType" name="SYBASE" minOccurs="0"/>
            <xs:element type="POSTGRESQLType" name="POSTGRESQL"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
...
     <xs:complexType name="POSTGRESQLType">
        <xs:sequence>
            <xs:element name="DB_UDC_RESTRICTION">
                <xs:simpleType>
                    <xs:restriction base="xs:integer">
                        <xs:enumeration value="1"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="DB_UDC_LIMIT">
                <xs:simpleType>
                    <xs:restriction base="xs:integer">
                        <xs:minInclusive value="1"/>
                        <xs:maxInclusive value="5000"/>
                    </xs:restriction>
```

```
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

...
```

Sample - Control List API for PostgreSQL/Pivotal Greenplum

We have added new CHECK_TYPE element to the XML output for Control List API: PostgreSQL Database Check.

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>
        <TECHNOLOGY_LIST>
          <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>
          <TECHNOLOGY>
            <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>
      </CONTROL>
    </CONTROL_LIST>
  </RESPONSE>
</CONTROL_LIST_OUTPUT>
```

Sample - Posture API

We've added support for PostgreSQL technologies (PostgreSQL 9.x, PostgreSQL 10.x, PostgreSQL 11.x, and PostgreSQL 12.x) and Pivotal Greenplum technologies (Pivotal Greenplum 5.x and Pivotal Greenplum 6.x) for the UDC, and you'll see these technologies in the Posture API, when applicable. In this sample, you'll see an instance with PostgreSQL 10.x.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=list&policy_id=3118495&details=All&host_ids=4471830"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/"
```

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>2020-10-15T20:39:08Z</DATETIME>
    <INFO_LIST>
      <INFO>
        <ID>10606254</ID>
        <HOST_ID>4471830</HOST_ID>
        <CONTROL_ID>101335</CONTROL_ID>
        <TECHNOLOGY_ID>143</TECHNOLOGY_ID>
        <INSTANCE>PostgreSQL 10.x:1:5432:postgres</INSTANCE>
        <STATUS>Passed</STATUS>
        <POSTURE_MODIFIED_DATE>2020-10-
15T20:35:31Z</POSTURE_MODIFIED_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>2020-10-15T20:35:31Z</FIRST_PASS_DATE>
        <LAST_PASS_DATE>2020-10-15T20:35:31Z</LAST_PASS_DATE>
        <EVIDENCE>
          <BOOLEAN_EXPR><![CDATA[(((:dp_1 in #fv_1 or :dp_1 matches $tp_1
)) and :dp_1 matches $tp_1)]]></BOOLEAN_EXPR>
```

```
            <DPV_LIST>
              <DPV lastUpdated="2020-10-15T20:09:53Z">
                <LABEL>:dp_1</LABEL>
                <V>
                  <H>
                    <C><![CDATA[name]]></C>
                    <C><![CDATA[setting]]></C>
                  </H>
                  <R>
                    <C><![CDATA[log_min_duration_statement]]></C>
                    <C><![CDATA[-1]]></C>
                  </R>
                </V>
              </DPV>
            </DPV_LIST>
          </EVIDENCE>
        </INFO>
      </INFO_LIST>

...

        <TECHNOLOGY_LIST>
          <TECHNOLOGY>
            <ID>143</ID>
            <NAME><![CDATA[PostgreSQL 10.x]]></NAME>
          </TECHNOLOGY>
        </TECHNOLOGY_LIST>
        <DPD_LIST>
          <DPD>
            <LABEL>:dp_1</LABEL>
            <DESC><![CDATA[Description]]></DESC>
          </DPD>
        </DPD_LIST>
        <TP_LIST>
          <TP>
            <LABEL>$tp_1</LABEL>
            <V><![CDATA[.*]]></V>
          </TP>
        </TP_LIST>
        <FV_LIST>
          <FV>
            <LABEL>#fv_1</LABEL>
            <V><![CDATA[Set status to PASS if no data found (No data
found)]]></V>
          </FV>
        </FV_LIST>
      </GLOSSARY>
    </RESPONSE>
</POSTURE_INFO_LIST_OUTPUT>
```

Sample - Policy Export API

We've added support for PostgreSQL technologies (PostgreSQL 9.x, PostgreSQL 10.x, PostgreSQL 11.x, and PostgreSQL 12.x) and Pivotal Greenplum technologies (Pivotal Greenplum 5.x and Pivotal Greenplum 6.x) for the UDC, and you'll see these technologies in the Policy Export API, when applicable.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?action=export
&id=3118495&show_user_controls=1"
```

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>2020-10-15T17:08:54Z</DATETIME>
<POLICY>
    <TITLE><![CDATA[prePostGreSQL_GreenPlum_DBUDC_Policy]]></TITLE>
    <EXPORTED><![CDATA[2020-10-15T17:08:54Z]]></EXPORTED>
    <COVER_PAGE><![CDATA[]]></COVER_PAGE>
    <STATUS><![CDATA[active]]></STATUS>
    <TECHNOLOGIES total="6">
        <TECHNOLOGY>
            <ID>114</ID>
            <NAME>PostgreSQL 9.x</NAME>
        </TECHNOLOGY>
        <TECHNOLOGY>
            <ID>143</ID>
            <NAME>PostgreSQL 10.x</NAME>
        </TECHNOLOGY>
        <TECHNOLOGY>
            <ID>192</ID>
            <NAME>PostgreSQL 11.x</NAME>
        </TECHNOLOGY>
        <TECHNOLOGY>
            <ID>201</ID>
            <NAME>Pivotal Greenplum 5.x</NAME>
        </TECHNOLOGY>
        <TECHNOLOGY>
            <ID>228</ID>
            <NAME>PostgreSQL 12.x</NAME>
        </TECHNOLOGY>
        <TECHNOLOGY>
            <ID>230</ID>
```

```
            <NAME>Pivotal Greenplum 6.x</NAME>
        </TECHNOLOGY>
    </TECHNOLOGIES>
    <SECTIONS total="1">
        <SECTION>
            <NUMBER>1</NUMBER>
            <HEADING><![CDATA[Untitled]]></HEADING>
            <CONTROLS total="1">
                <USER_DEFINED_CONTROL>
                    <ID>101335</ID>
                    <UDC_ID>b9626358-8450-c78d-83a4-543899ad1366</UDC_ID>
                    <CHECK_TYPE>PostgreSQL Database Check</CHECK_TYPE>
                    <CATEGORY>
                        <ID>3</ID>
                      <NAME><![CDATA[Access Control Requirements]]></NAME>
                    </CATEGORY>
                    <SUB_CATEGORY>
                        <ID>1010</ID>
                        <NAME><![CDATA[Account Creation/User
Management]]></NAME>
                    </SUB_CATEGORY>

<STATEMENT><![CDATA[prePostGreSQL_selectStatement]]></STATEMENT>
                    <CRITICALITY>
                        <LABEL><![CDATA[URGENT]]></LABEL>
                        <VALUE>5</VALUE>
                    </CRITICALITY>
                    <COMMENT><![CDATA[comments]]></COMMENT>
                    <IGNORE_ERROR>0</IGNORE_ERROR>
                    <ERROR_SET_STATUS></ERROR_SET_STATUS>
                    <TECHNOLOGIES total="6">
                        <TECHNOLOGY>
                            <ID>114</ID>
                            <NAME>PostgreSQL 9.x</NAME>

<EVALUATE><CTRL><AND><OR><DP><K>custom.pgsql_query.2580143</K><OP>xre</OP
><CD>matches</CD><FV set="1">No data
found</FV><DT>5</DT><V><![CDATA[.*]]></V><DBCOL><![CDATA[]]></DBCOL></DP>
</OR></AND></CTRL></EVALUATE>
                            <RATIONALE><![CDATA[Rationale]]></RATIONALE>
                            <REMEDIATION><![CDATA[Reme]]></REMEDIATION>
                            <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>143</ID>
```

```
                            <NAME>PostgreSQL 10.x</NAME>

<EVALUATE><CTRL><AND><OR><DP><K>custom.pgsql_query.2580143</K><OP>xre</OP
><CD>matches</CD><FV set="1">No data
found</FV><DT>5</DT><V><![CDATA[.*]]></V><DBCOL><![CDATA[]]></DBCOL></DP>
</OR></AND></CTRL></EVALUATE>
                            <RATIONALE><![CDATA[Rationale]]></RATIONALE>
                            <REMEDIATION><![CDATA[Reme]]></REMEDIATION>
                            <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>

<EVALUATE><CTRL><AND><OR><DP><K>custom.pgsql_query.2580143</K><OP>xre</OP
><CD>matches</CD><FV set="1">No data
found</FV><DT>5</DT><V><![CDATA[.*]]></V><DBCOL><![CDATA[]]></DBCOL></DP>
</OR></AND></CTRL></EVALUATE>
                            <RATIONALE><![CDATA[Rationale]]></RATIONALE>
                            <REMEDIATION><![CDATA[Reme]]></REMEDIATION>
                            <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>

<EVALUATE><CTRL><AND><OR><DP><K>custom.pgsql_query.2580143</K><OP>xre</OP
><CD>matches</CD><FV set="1">No data
found</FV><DT>5</DT><V><![CDATA[.*]]></V><DBCOL><![CDATA[]]></DBCOL></DP>
</OR></AND></CTRL></EVALUATE>
                            <RATIONALE><![CDATA[Rationale]]></RATIONALE>
                            <REMEDIATION><![CDATA[Reme]]></REMEDIATION>
                            <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>
```

```
<EVALUATE><CTRL><AND><OR><DP><K>custom.pgsql_query.2580143</K><OP>xre</OP
><CD>matches</CD><FV set="1">No data
found</FV><DT>5</DT><V><![CDATA[.*]]></V><DBCOL><![CDATA[]]></DBCOL></DP>
</OR></AND></CTRL></EVALUATE>
                            <RATIONALE><![CDATA[Rationale]]></RATIONALE>
                            <REMEDIATION><![CDATA[Reme]]></REMEDIATION>
                            <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>

<EVALUATE><CTRL><AND><OR><DP><K>custom.pgsql_query.2580143</K><OP>xre</OP
><CD>matches</CD><FV set="1">No data
found</FV><DT>5</DT><V><![CDATA[.*]]></V><DBCOL><![CDATA[]]></DBCOL></DP>
</OR></AND></CTRL></EVALUATE>
                            <RATIONALE><![CDATA[Rationale]]></RATIONALE>
                            <REMEDIATION><![CDATA[Reme]]></REMEDIATION>
                            <DB_QUERY><![CDATA[select name, setting from
pg_catalog.pg_settings where
name='log_min_duration_statement']]></DB_QUERY>

<DESCRIPTION><![CDATA[Description]]></DESCRIPTION>
                        </TECHNOLOGY>
                    </TECHNOLOGIES>
                    <REFERENCE_LIST/>
                </USER_DEFINED_CONTROL>
            </CONTROLS>
        </SECTION>
    </SECTIONS>
</POLICY>
  </RESPONSE>
</POLICY_EXPORT_OUTPUT>
```

Schema update (ImportableControl.xsd):

The ImportableControl.xsd schema is used when importing and exporting controls. We added the enumeration value PostgreSQL Database Check to the schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
...
<xs:element name="CHECK_TYPE">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="Registry Key Existence" />
                <xs:enumeration value="Registry Value Existence" />
                <xs:enumeration value="Registry Value Content Check" />
                <xs:enumeration value="Registry Permission" />
               <xs:enumeration value="Window File/Directory Existence" />
              <xs:enumeration value="Window File/Directory Permission" />
               <xs:enumeration value="Unix File/Directory Permission" />
               <xs:enumeration value="Unix File Content Check" />
               <xs:enumeration value="Unix File/Directory Existence" />
               <xs:enumeration value="Window File Integrity Check" />
               <xs:enumeration value="Unix File Integrity Check" />
               <xs:enumeration value="WMI Query Check" />
               <xs:enumeration value="Share Access Check" />
               <xs:enumeration value="Unix Directory Search Check" />
               <xs:enumeration value="Windows Directory Search Check" />
               <xs:enumeration value="Windows Group Membership Check" />
               <xs:enumeration value="Windows Directory Integrity Check"
/>
               <xs:enumeration value="Unix Directory Integrity Check" />
               <xs:enumeration value="MSSQL Database Check" />
               <xs:enumeration value="Oracle Database Check" />
               <xs:enumeration value="Sybase Database Check" />
               <xs:enumeration value="PostgreSQL Database Check" />
               <xs:enumeration value="Windows File Content Check" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
...
```