Qualys Cloud Platform (VM, PC) v10.x

API Release Notes

Version 10.11

May 13, 2021

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

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.

Posture Info API - CSV and XML Output Changes for Consistency

APIs affected	/api/2.0/fo/compliance/posture/info/
New or Updated API	Updated
DTD or XSD changes	Yes

For the Posture Info API, we added several new columns to the CSV output and a new tag to the XML output to make these formats more consistent. This way you'll get similar details in both formats.

New columns added to CSV format:

Evaluation Date - The date/time when compliance posture was last evaluated on the host.

Host ID - The unique host ID assigned to each host instance. You'll see the host ID in several Qualys (VM, PC) APIs, e.g. Host List API.

Instance - A single instance on the host. For example, the instance string may be "os" or a string like "oracle10:1:1521:ora10204u".

Asset ID - The unique asset ID assigned to each host asset in your subscription. You'll see the asset ID in several Asset Management APIs.

Posture Modified Date - The date/time when the compliance posture was last changed on the host.

Last Compliance Scan Date - The date/time when a compliance scan was last launched on the host.

Last Vuln Scan Date - The date/time when a vulnerability scan was last launched on the host.

New tag added to XML format:

<ASSET_ID> - The unique asset ID assigned to each host asset in your subscription. You'll see the asset ID in several Asset Management APIs.

Sample - Posture Info in CSV Format

This sample shows the new columns (in bold) in CSV format.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=list&policy_id=29476&details=All&output_format=csv"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/"
```

CSV Output:

```
----BEGIN_RESPONSE_BODY_CSV
"POLICY ID","DATETIME"
"29476","04/26/2021 16:20:09"

"ID","IP","OS","DNS Name","NetBios","Tracking Method","Control
ID","Control Statement","Criticality Label","Criticality
Value","Technology ID","Technology Name","Posture","Previous
Status","First Fail Date","Last Fail Date","First Pass Date","Last Pass
Date","Evaluation Date","Qualys Host ID","Posture
Evidence","Reference","Host ID","Instance","Asset ID","Posture Modified
Date","Last Compliance Scan Date","Last vuln Scan Date"
"96770","10.11.12.13","Windows Server 2008 R2 Enterprise 64 bit
Edition",,"SYS_10_11_12_13","IP","1502","Status of the 'Current Service
Pack Version' installed on the system","URGENT","5","21","Windows 2008
Server","Failed","Failed","01/28/2021 09:14:34","01/28/2021
09:14:34","N/A","N/A","01/28/2021 09:14:24",,"This String value <B>X</B>
indicates the current version of the <B>service pack</B> as defined within
the <B>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows
NT\CurrentVersion\CSDVersion</B> registry key.

======Expected Value(s)======
regular expression match
^Service Pack 1$

======Current Value(s) - Last updated: 01/15/2021 at 05:56:34 (GMT)======
Key not found","V-1073","686132","os","689027","01/28/2021
09:14:24","01/15/2021 05:56:34","N/A"
...
```

Sample - Posture Info in XML Format

This sample shows the new <ASSET_ID> tag (in bold) in XML format.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=list&policy_id=29476&details=All&output_format=xml"
"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>2021-04-26T16:16:03Z</DATETIME>
```

```
    <INFO_LIST>
      <INFO>
        <ID>96770</ID>
        <HOST_ID>686132</HOST_ID>
        <CONTROL_ID>1502</CONTROL_ID>
        <TECHNOLOGY_ID>21</TECHNOLOGY_ID>
        <INSTANCE></INSTANCE>
        <STATUS>Failed</STATUS>
        <POSTURE_MODIFIED_DATE>2021-01-
28T09:14:24Z</POSTURE_MODIFIED_DATE>
        <EVALUATION_DATE>2021-01-28T09:14:24Z</EVALUATION_DATE>
        <PREVIOUS_STATUS>Failed</PREVIOUS_STATUS>
        <FIRST_FAIL_DATE>2021-01-28T09:14:34Z</FIRST_FAIL_DATE>
        <LAST_FAIL_DATE>2021-01-28T09:14:34Z</LAST_FAIL_DATE>
        <FIRST_PASS_DATE>N/A</FIRST_PASS_DATE>
        <LAST_PASS_DATE>N/A</LAST_PASS_DATE>
        <EVIDENCE>
         <BOOLEAN_EXPR><![CDATA[(:dp_75 in #fv_2 or :dp_75 matches $tp_13
)]]></BOOLEAN_EXPR>
          <DPV_LIST>
            <DPV lastUpdated="2021-01-15T05:56:34Z">
              <LABEL>:dp_75</LABEL>
              <V><![CDATA[314159265358979]]></V>
            </DPV>
          </DPV_LIST>
        </EVIDENCE>
      </INFO>
...

      <HOST_LIST>
        <HOST>
          <ID>686132</ID>
          <IP network_id="0">10.11.12.13</IP>
          <TRACKING_METHOD>IP</TRACKING_METHOD>
          <NETBIOS><![CDATA[SYS_10_11_12_13]]></NETBIOS>
          <OS><![CDATA[Windows Server 2008 R2 Enterprise 64 bit
Edition]]></OS>
          <ASSET_ID>689027</ASSET_ID>
          <LAST_COMPLIANCE_SCAN_DATETIME>2021-01-
15T05:56:34Z</LAST_COMPLIANCE_SCAN_DATETIME>
          <PERCENTAGE><![CDATA[48.01% (169 of 352)]]></PERCENTAGE>
        </HOST>
...
```

DTD update:

DTD: <platform>/api/2.0/fo/compliance/posture/info/posture_info_list_output.dtd

We added ASSET_ID to the Posture Info List Output DTD.

```
<!-- QUALYS POSTURE_INFO_LIST_OUTPUT DTD -->
<!-- $Revision$ -->
<!ELEMENT POSTURE_INFO_LIST_OUTPUT (REQUEST?,RESPONSE)>

...

<!ELEMENT GLOSSARY (USER_LIST?, HOST_LIST, CONTROL_LIST?,
TECHNOLOGY_LIST?, DPD_LIST?, TP_LIST?, FV_LIST?, TM_LIST?)>
<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>

<!ELEMENT HOST_LIST (HOST+)>
<!ELEMENT HOST (ID, IP, TRACKING_METHOD, DNS?, DNS_DATA?, NETBIOS?, OS?,
OS_CPE?, QG_HOSTID?, ASSET_ID?, LAST_VULN_SCAN_DATETIME?,
LAST_COMPLIANCE_SCAN_DATETIME?, PERCENTAGE?)>
<!ELEMENT TRACKING_METHOD (#PCDATA)>
<!ELEMENT IP (#PCDATA)>
<!ATTLIST IP network_id  CDATA  #IMPLIED>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT DNS_DATA (HOSTNAME?, DOMAIN?, FQDN?)>
<!ELEMENT HOSTNAME (#PCDATA)>
<!ELEMENT DOMAIN (#PCDATA)>
<!ELEMENT FQDN (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT OS (#PCDATA)>
<!ELEMENT OS_CPE (#PCDATA)>
<!ELEMENT QG_HOSTID (#PCDATA)>
<!ELEMENT ASSET_ID (#PCDATA)>
<!ELEMENT LAST_VULN_SCAN_DATETIME (#PCDATA)>
<!ELEMENT LAST_COMPLIANCE_SCAN_DATETIME (#PCDATA)>
<!ELEMENT PERCENTAGE (#PCDATA)>

...
```

New Authentication Support for Neo4j

APIs affected	/api/2.0/fo/auth/
New or Updated API	Updated
DTD or XSD changes	Yes
APIs affected	/api/2.0/fo/auth/neo4j/
New or Updated API	New
DTD or XSD changes	New

Neo4j authentication is now supported for compliance scans. The new Neo4j Authentication API (api/2.0/fo/auth/neo4j/) lets you list, create, update, and delete Neo4j authentication records. User permissions for this API are the same as other authentication record APIs.

API Sample - List All Record Types

Use the Authentication Record List API (/api/2.0/fo/auth/?action=list) to list records. You'll see Neo4j record IDs in the output when you have Neo4j 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/"
```

Response:

```
<?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>2021-05-12T12:20:45Z</DATETIME>
    <AUTH_RECORDS>
      <AUTH_UNIX_IDS>
        <ID_SET>
          <ID>604909</ID>
          <ID_RANGE>980093-980094</ID_RANGE>
          <ID>1033086</ID>
          <ID_RANGE>1085432-1085433</ID_RANGE>
          <ID>4574304</ID>
        </ID_SET>
      </AUTH_UNIX_IDS>
      <AUTH_WINDOWS_IDS>
        <ID_SET>
          <ID>4476044</ID>
          <ID>4493943</ID>
```

```
            <ID>4574305</ID>
            <ID>4672942</ID>
          </ID_SET>
        </AUTH_WINDOWS_IDS>
        <AUTH_NEO4J_IDS>
          <ID_SET>
            <ID>4716287</ID>
            <ID>4725193</ID>
            <ID_RANGE>4734508-4734509</ID_RANGE>
          </ID_SET>
        </AUTH_NEO4J_IDS>
      </AUTH_RECORDS>
    </RESPONSE>
</AUTH_RECORDS_OUTPUT>
```

Updated DTD

DTD: <platform>/api/2.0/fo/auth/auth_records.dtd

The element AUTH_NEO4J_IDS was added to identify Neo4j record IDs.

```
<!-- QUALYS AUTH_RECORDS_OUTPUT DTD -->
<!-- $Revision$ -->
<!ELEMENT AUTH_RECORDS_OUTPUT (REQUEST?, RESPONSE)>
...
<!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? )>
...
<!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_ORACLE_HTTP_SERVER_IDS (ID_SET)>
<!ELEMENT AUTH_GREENPLUM_IDS (ID_SET)>
<!ELEMENT AUTH_MICROSOFT_SHAREPOINT_IDS (ID_SET)>
<!ELEMENT AUTH_KUBERNETES_IDS (ID_SET)>
<!ELEMENT AUTH_SAPIQ_IDS (ID_SET)>
<!ELEMENT AUTH_SAP_HANA_IDS (ID_SET)>
<!ELEMENT AUTH_NEO4J_IDS (ID_SET)>
...
```

List Neo4j Records

Use the new Neo4j Authentication Record List API (/api/2.0/fo/auth/neo4j/?action=list) to list only Neo4j records.

Input Parameters

Parameter	Description
action=list	(Required) Specify list (using GET or POST) to list records.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
"action=list" "https://qualysapi.qualys.com/api/2.0/fo/auth/neo4j/"
```

Response:

```
<?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_outpu
t.dtd">
<AUTH_NEO4J_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2021-03-15T11:48:54Z</DATETIME>
        <AUTH_NEO4J_LIST>
          <AUTH_NEO4J>
                <ID>101428</ID>
                <TITLE>
                    <![CDATA[Neo4j]]>
                </TITLE>
                <USERNAME>
                    <![CDATA[root1]]>
                </USERNAME>
                <DATABASE>
                    <![CDATA[ABC]]>
                </DATABASE>
                <PORT>1255</PORT>
                <SSL_VERIFY>
                    <![CDATA[1]]>
                </SSL_VERIFY>
                <HOSTS>
                    <HOST>
                        <![CDATA[test.domain.com]]>
                    </HOST>
                </HOSTS>
                <IP_SET>
                    <IP>10.115.95.104</IP>
                </IP_SET>
                <IS_CDB>/neo4j/conf1</IS_CDB>
```

```
                    <CONF_PATH>
                        <![CDATA[/etc/var/www/neo4j1]]>
                    </CONF_PATH>
                    <BASE_PATH>
                        <![CDATA[/neo4j/conf1]]>
                    </BASE_PATH>
                    <VERSION>
                        <![CDATA[neo4j 3.x]]>
                    </VERSION>
                    <AUTO_PATH>
                        <![CDATA[0]]>
                    </AUTO_PATH>
                    <LOGIN_TYPE>
                        <![CDATA[basic]]>
                    </LOGIN_TYPE>
                    <CREATED>
                        <DATETIME>2021-03-15T10:18:21Z</DATETIME>
                        <BY>joe_user</BY>
                    </CREATED>
                    <LAST_MODIFIED>
                        <DATETIME>2021-03-15T10:27:07Z</DATETIME>
                    </LAST_MODIFIED>
                </AUTH_NEO4J>
            </AUTH_NEO4J_LIST>
        </RESPONSE>
    </AUTH_NEO4J_LIST_OUTPUT>
```

New DTD:

DTD: <platform>/api/2.0/fo/auth/neo4j/fo/auth/neo4j/auth_neo4j_list_output.dtd

```
<!ELEMENT AUTH_NEO4J_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_NEO4J_LIST|ID_SET)?, WARNING_LIST?,
GLOSSARY?)>
<!ELEMENT AUTH_NEO4J_LIST (AUTH_NEO4J+)>
```

```
<!ELEMENT AUTH_NEO4J (ID, TITLE,
USERNAME,DATABASE,PORT,SSL_VERIFY?,HOSTS?,IP_SET?,IS_CDB?,UNIX_CONF_PATH?
,UNIX_BASE_PATH?,VERSION?,AUTO_PATH?,PASSWORD_ENCRYPTION?,LOGIN_TYPE?,DIG
ITAL_VAULT?,NETWORK_ID?,CREATED,LAST_MODIFIED,COMMENTS?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT DATABASE (#PCDATA)>
<!ELEMENT PORT (#PCDATA)>
<!ELEMENT SSL_VERIFY (#PCDATA)>
<!ELEMENT HOSTS (HOST+)>
<!ELEMENT HOST (#PCDATA)>
<!ELEMENT IP_SET (IP|IP_RANGE)+>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>
<!ELEMENT IS_CDB (#PCDATA)>
<!ELEMENT UNIX_CONF_PATH (#PCDATA)>
<!ELEMENT UNIX_BASE_PATH (#PCDATA)>
<!ELEMENT VERSION (#PCDATA)>
<!ELEMENT AUTO_PATH (#PCDATA)>
<!ELEMENT PASSWORD_ENCRYPTION (#PCDATA)>

<!ELEMENT LOGIN_TYPE (#PCDATA)>
<!ELEMENT DIGITAL_VAULT (DIGITAL_VAULT_ID, DIGITAL_VAULT_TYPE,
DIGITAL_VAULT_TITLE, VAULT_USERNAME?, VAULT_FOLDER?, VAULT_FILE?,
VAULT_SECRET_NAME?, VAULT_SYSTEM_NAME?, VAULT_NS_TYPE?, VAULT_NS_NAME?,
VAULT_SECRET_KV_PATH?, VAULT_SECRET_KV_NAME?, VAULT_SECRET_KV_KEY?,
VAULT_SERVICE_TYPE?)>
<!ELEMENT DIGITAL_VAULT_ID (#PCDATA)>
<!ELEMENT DIGITAL_VAULT_TYPE (#PCDATA)>
<!ELEMENT DIGITAL_VAULT_TITLE (#PCDATA)>
<!ELEMENT VAULT_USERNAME (#PCDATA)>
<!ELEMENT VAULT_FOLDER (#PCDATA)>
<!ELEMENT VAULT_FILE (#PCDATA)>
<!ELEMENT VAULT_SECRET_NAME (#PCDATA)>
<!ELEMENT VAULT_SYSTEM_NAME (#PCDATA)>
<!ELEMENT VAULT_NS_TYPE (#PCDATA)>
<!ELEMENT VAULT_NS_NAME (#PCDATA)>
<!ELEMENT VAULT_SECRET_KV_PATH (#PCDATA)>
<!ELEMENT VAULT_SECRET_KV_NAME (#PCDATA)>
<!ELEMENT VAULT_SECRET_KV_KEY (#PCDATA)>
<!ELEMENT VAULT_SERVICE_TYPE (#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 -->
```

Create/Update Neo4j Records

Use these parameters to create or update a Neo4j authentication record.

Input Parameters

Parameter	Description
action={action}	(Required) Specify create, update, delete (using POST) or list (using GET or POST).
ids={value}	(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.
title={value}	(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).
ips={value}	(Required to create record) 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 IP list for an existing record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
remove_ips={value}	(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.
database={value}	(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.

port={value}	(Required to create record, optional to update record) The port number assigned to the database instance to be scanned.
username={value}	(Required to create record, optional to update record) The username to be used for authentication to Neo4j.
login_type={**basic**|vault}	(Optional) The login type is basic by default. You can choose vault (for vault based authentication).
password={value}	(Required to create record) When login_type=basic, specify the password to be used for authentication to Neo4j.
ssl_verify={0|1}	(Optional to create or update record, and valid for server that supports SSL) Specify 1 for a complete SSL certificate validation. - 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. - 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.
hosts={value}	(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.
neo4j_version={value}	(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.
unix_base_path={value}	(Optional) The base path for Neo4j on your Unix hosts. Sample value: /opt/neo4j-enterprise-3.5.16/ Instead of specifying the path information, you can choose to auto discover the base and configuration paths by specifying neo4j_auto_path=1.

unix_conf_path={value}	(Optional) The path to the Neo4j configuration file on your Unix hosts. Sample value: /opt/neo4j-enterprise-3.5.16/conf/neo4j.conf 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.
neo4j_auto_path={**0**|1}	(Optional) When unspecified or neo4j_auto_path=0 (false), we will not use auto discovery to find the base and configuration paths for Neo4j on your Unix hosts. Use the unix_base_path and unix_conf_path input parameters to specify path information. When neo4j_auto_path=1 (true) we will auto discover the base and configuration paths for Neo4j on your Unix hosts.

Example: Create Neo4j Record

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
"action=create&title=neo4j-recordAuth
Record&username=root&password=root1&database=graph.db&port=7687&ips=1.1.1
4&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>
```

Example: Update Neo4j Record

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
"action=update&title=Neo4j Auth Record
&username=root&password=root1&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/neo4jl/"
```

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>
```

Delete Neo4j Records

Use these parameters to delete authentication records.

Input Parameters

Parameter	Description
action=delete	(Required) POST method may be used.
ids={value}	(Required) Neo4j authentication record IDs for the records you want to delete. Multiple records are comma separated.

Example: 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>
```

New Authentication Support for Network SSH

APIs affected	/api/2.0/fo/auth/
New or Updated API	Updated
DTD or XSD changes	Yes
APIs affected	/api/2.0/fo/auth/network_ssh/
New or Updated API	New
DTD or XSD changes	New

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.

With this release, we've extended functionality to authenticate network devices (such as Cisco and Checkpoint Firewall) using SSH2 authentication format. Previously only Unix devices were supported to use SSH2 authentication format.

Network SSH authentication record can be used in place of the Cisco and Checkpoint Firewall authentication records. This new authentication record has all the same functionality as the previous Cisco and Checkpoint Firewall records along with newly added 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 existing expert_password field (for Checkpoint Firewall sub-type) and enable_password field (for Cisco sub-type).

List All Authentication Records

Use the Authentication Record List API (/api/2.0/fo/auth/?action=list) to list authentication records for all types. You'll see <AUTH_NETWORK_SSH_IDS> in the output when you have Network SSH records in your account.

API request:

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

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_RECORDS_OUTPUT SYSTEM
"http://qualysapi.qualys.com/api/2.0/fo/auth/auth_records.dtd">
<AUTH_RECORDS_OUTPUT>
    <RESPONSE>
        <DATETIME>2021-04-21T06:06:22Z</DATETIME>
        <AUTH_RECORDS>
```

```
            <AUTH_UNIX_IDS>
                <ID_SET>
                    <ID_RANGE>99178-99179</ID_RANGE>
                    <ID>99207</ID>
                    <ID_RANGE>101550-101551</ID_RANGE>
                </ID_SET>
            </AUTH_UNIX_IDS>
            <AUTH_AZURE_MS_SQL_IDS>
                <ID_SET>
                    <ID>101466</ID>
                </ID_SET>
            </AUTH_AZURE_MS_SQL_IDS>
            <AUTH_NETWORK_SSH_IDS>
                <ID_SET>
                    <ID>102419</ID>
                    <ID>102440</ID>
                    <ID>102448</ID>
                    <ID>102450</ID>
                </ID_SET>
            </AUTH_NETWORK_SSH_IDS>
        </AUTH_RECORDS>
    </RESPONSE>
</AUTH_RECORDS_OUTPUT>
```

Updated DTD:

DTD: <platform API server>/api/2.0/fo/auth/auth_records.dtd

The element AUTH_NETWORK_SSH_IDS has been added to identify Network SSH record IDs.

```
<!-- QUALYS AUTH_RECORDS_OUTPUT DTD -->
<!-- $Revision$ -->
<!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_AZURE_MS_SQL_IDS?,
AUTH_NETWORK_SSH_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)>
<!ELEMENT AUTH_MONGODB_IDS (ID_SET)>
<!ELEMENT AUTH_PALO_ALTO_FIREWALL_IDS (ID_SET)>
<!ELEMENT AUTH_VCENTER_IDS (ID_SET)>
<!ELEMENT AUTH_JBOSS_IDS (ID_SET)>
<!ELEMENT AUTH_MARIADB_IDS (ID_SET)>
<!ELEMENT AUTH_INFORMIXDB_IDS (ID_SET)>
<!ELEMENT AUTH_MS_EXCHANGE_IDS (ID_SET)>
<!ELEMENT AUTH_ORACLE_HTTP_SERVER_IDS (ID_SET)>
<!ELEMENT AUTH_GREENPLUM_IDS (ID_SET)>
<!ELEMENT AUTH_MICROSOFT_SHAREPOINT_IDS (ID_SET)>
<!ELEMENT AUTH_KUBERNETES_IDS (ID_SET)>
<!ELEMENT AUTH_SAPIQ_IDS (ID_SET)>
<!ELEMENT AUTH_SAP_HANA_IDS (ID_SET)>
<!ELEMENT AUTH_AZURE_MS_SQL_IDS (ID_SET)>
<!ELEMENT AUTH_NETWORK_SSH_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 -->
```

List Network SSH Authentication Records

Use these parameters to list Network SSH authentication records.

Parameter	Description
action={action}	(Required) Specify list (using GET or POST) to list records.
details={value}	(Optional) Default value is Basic. You can choose from None, Basic, and All.
ids={value}	(Optional) Network SSH auth record IDs to list. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.

Sample - List Network SSH Authentication Records

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"https://qualysapi.qualys.com/api/2.0/fo/auth/network_ssh/?action=list&id
s=102419"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_NETWORK_SSH_LIST_OUTPUT SYSTEM
"http://qualysapi.qualys.com/api/2.0/fo/auth/network_ssh/dtd/auth_list_ou
tput.dtd">
<AUTH_NETWORK_SSH_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2021-04-21T06:01:42Z</DATETIME>
        <AUTH_NETWORK_SSH_LIST>
            <AUTH_NETWORK_SSH>
                <ID>102419</ID>
                <TITLE>
                    <![CDATA[nn2]]>
                </TITLE>
                <USERNAME>
                    <![CDATA[abc]]>
                </USERNAME>
                <PORT>17, 122, 270</PORT>
                <IP_SET>
```

```
    <IP>10.10.0.3</IP>
</IP_SET>
<CLEARTEXT_PASSWORD>1</CLEARTEXT_PASSWORD>
<TARGET_TYPE>
    <![CDATA[ArubaOS]]>
</TARGET_TYPE>
<PRIVATE_KEY_CERTIFICATE_LIST>
    <PRIVATE_KEY_CERTIFICATE>
        <ID>18028</ID>
        <PRIVATE_KEY_INFO type="vault">
            <DIGITAL_VAULT>
                <DIGITAL_VAULT_ID>
                    <![CDATA[41022]]>
                </DIGITAL_VAULT_ID>
                <DIGITAL_VAULT_TYPE>
                    <![CDATA[CA PAM]]>
                </DIGITAL_VAULT_TYPE>
                <DIGITAL_VAULT_TITLE>
                    <![CDATA[CA pam]]>
                </DIGITAL_VAULT_TITLE>
                <VAULT_DEVICE_NAME>
                    <![CDATA[hq_device]]>
                </VAULT_DEVICE_NAME>
                <VAULT_DEVICE_HOST>
                    <![CDATA[]]>
                </VAULT_DEVICE_HOST>
                <VAULT_APP_NAME>
                    <![CDATA[APP_NAME]]>
                </VAULT_APP_NAME>
            </DIGITAL_VAULT>
        </PRIVATE_KEY_INFO>
        <PASSPHRASE_INFO type="vault">
            <DIGITAL_VAULT>
                <DIGITAL_VAULT_ID>
                    <![CDATA[41022]]>
                </DIGITAL_VAULT_ID>
                <DIGITAL_VAULT_TYPE>
                    <![CDATA[CA PAM]]>
                </DIGITAL_VAULT_TYPE>
                <DIGITAL_VAULT_TITLE>
                    <![CDATA[CA pam]]>
                </DIGITAL_VAULT_TITLE>
                <VAULT_DEVICE_NAME>
                    <![CDATA[hq_device]]>
                </VAULT_DEVICE_NAME>
                <VAULT_DEVICE_HOST>
                    <![CDATA[]]>
                </VAULT_DEVICE_HOST>
                <VAULT_APP_NAME>
```

```
                                        <![CDATA[APP_NAME]]>
                                    </VAULT_APP_NAME>
                                </DIGITAL_VAULT>
                            </PASSPHRASE_INFO>
                        </PRIVATE_KEY_CERTIFICATE>
                    </PRIVATE_KEY_CERTIFICATE_LIST>
                    <NETWORK_ID>0</NETWORK_ID>
                    <CREATED>
                        <DATETIME>2021-04-12T06:58:56Z</DATETIME>
                        <BY>new_gc</BY>
                    </CREATED>
                    <LAST_MODIFIED>
                        <DATETIME>2021-04-21T05:06:19Z</DATETIME>
                    </LAST_MODIFIED>
                    <COMMENTS>
                        <![CDATA[new auth record]]>
                    </COMMENTS>
                </AUTH_NETWORK_SSH>
            </AUTH_NETWORK_SSH_LIST>
        </RESPONSE>
    </AUTH_NETWORK_SSH_LIST_OUTPUT>
```

New DTD:

DTD: <platform API server>/api/2.0/fo/auth/network_ssh/dtd/auth_list_output.dtd

```
<!-- QUALYS NETWORK_SSH_LIST_OUTPUT DTD -->
<!-- $Revision$ -->
<!ELEMENT AUTH_NETWORK_SSH_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_NETWORK_SSH_LIST|ID_SET)?,
WARNING_LIST?, GLOSSARY?)>
<!ELEMENT AUTH_NETWORK_SSH_LIST (AUTH_NETWORK_SSH+)>

<!ELEMENT AUTH_NETWORK_SSH (ID, TITLE, USERNAME, PORT?, IP_SET?,
LOGIN_TYPE?, DIGITAL_VAULT?, CLEARTEXT_PASSWORD?, PASSWORD2_INFO?,
TARGET_TYPE?, ((RSA_PRIVATE_KEY?,
```

```
DSA_PRIVATE_KEY?)|PRIVATE_KEY_CERTIFICATE_LIST?), NETWORK_ID?, CREATED,
LAST_MODIFIED, COMMENTS?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT CLEARTEXT_PASSWORD (#PCDATA)>
<!ELEMENT TARGET_TYPE (#PCDATA)>
<!ELEMENT RSA_PRIVATE_KEY EMPTY>
<!ELEMENT DSA_PRIVATE_KEY EMPTY>
<!ELEMENT PRIVATE_KEY_CERTIFICATE_LIST (PRIVATE_KEY_CERTIFICATE)*>
<!ELEMENT PORT (#PCDATA)>

<!ELEMENT IP_SET (IP|IP_RANGE)+>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>

<!ELEMENT LOGIN_TYPE (#PCDATA)>
<!ELEMENT PASSWORD2_INFO (DIGITAL_VAULT?)>
<!ATTLIST PASSWORD2_INFO type (basic|vault) "basic">
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT CREATED (DATETIME, BY)>
<!ELEMENT BY (#PCDATA)>
<!ELEMENT LAST_MODIFIED (DATETIME)>
<!ELEMENT COMMENTS (#PCDATA)>

<!-- Private key contents will never be rendered -->
<!ELEMENT PRIVATE_KEY_CERTIFICATE (ID, PRIVATE_KEY_INFO, PASSPHRASE_INFO,
CERTIFICATE?)+>
<!ELEMENT PRIVATE_KEY_INFO (PRIVATE_KEY|DIGITAL_VAULT)>
<!ATTLIST PRIVATE_KEY_INFO type (basic|vault) "basic">
<!-- Private key/Certificate contents will never be rendered -->
<!ELEMENT PRIVATE_KEY EMPTY>
<!ATTLIST PRIVATE_KEY type (rsa|dsa|ecdsa|ed25519) #REQUIRED>

<!ELEMENT PASSPHRASE_INFO (DIGITAL_VAULT?)>
<!ATTLIST PASSPHRASE_INFO type (basic|vault) "basic">

<!ELEMENT CERTIFICATE EMPTY>
<!ATTLIST CERTIFICATE type (x.509|openssh) #REQUIRED>
<!ELEMENT DIGITAL_VAULT (DIGITAL_VAULT_ID, DIGITAL_VAULT_TYPE,
DIGITAL_VAULT_TITLE, VAULT_USERNAME?, VAULT_FOLDER?, VAULT_FILE?,
VAULT_SECRET_NAME?, VAULT_SYSTEM_NAME?, VAULT_EP_NAME?, VAULT_EP_TYPE?,
VAULT_EP_CONT?, VAULT_NS_TYPE?, VAULT_NS_NAME?, VAULT_ACCOUNT_NAME?,
VAULT_AUTHORIZATION_NAME?, VAULT_TARGET_NAME?, VAULT_SECRET_KV_PATH?,
VAULT_SECRET_KV_NAME?, VAULT_SECRET_KV_KEY?, VAULT_DEVICE_NAME?,
VAULT_DEVICE_HOST?, VAULT_APP_NAME?, VAULT_SERVICE_TYPE?)>
<!ELEMENT DIGITAL_VAULT_ID (#PCDATA)>
<!ELEMENT DIGITAL_VAULT_TYPE (#PCDATA)>
<!ELEMENT DIGITAL_VAULT_TITLE (#PCDATA)>
```

```
<!ELEMENT VAULT_USERNAME (#PCDATA)>
<!ELEMENT VAULT_FOLDER (#PCDATA)>
<!ELEMENT VAULT_FILE (#PCDATA)>
<!ELEMENT VAULT_SECRET_NAME (#PCDATA)>
<!ELEMENT VAULT_SYSTEM_NAME (#PCDATA)>
<!ELEMENT VAULT_EP_NAME (#PCDATA)>
<!ELEMENT VAULT_EP_TYPE (#PCDATA)>
<!ELEMENT VAULT_EP_CONT (#PCDATA)>
<!ELEMENT VAULT_NS_TYPE (#PCDATA)>
<!ELEMENT VAULT_NS_NAME (#PCDATA)>
<!ELEMENT VAULT_ACCOUNT_NAME (#PCDATA)>
<!ELEMENT VAULT_AUTHORIZATION_NAME (#PCDATA)>
<!ELEMENT VAULT_TARGET_NAME (#PCDATA)>
<!ELEMENT VAULT_SECRET_KV_PATH (#PCDATA)>
<!ELEMENT VAULT_SECRET_KV_NAME (#PCDATA)>
<!ELEMENT VAULT_SECRET_KV_KEY (#PCDATA)>
<!ELEMENT VAULT_DEVICE_NAME (#PCDATA)>
<!ELEMENT VAULT_DEVICE_HOST (#PCDATA)>
<!ELEMENT VAULT_APP_NAME (#PCDATA)>
<!ELEMENT VAULT_SERVICE_TYPE (#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 -->
```

Create/Update Network SSH Authentication Records

Use these parameters to create or update Network SSH authentication records.

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

Parameter	Description
id={value}	(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.
title={value}	(Required to create record) A title for the record. The title must be unique. Maximum 255 characters (ascii).
comments={value}	(Optional to create or update record) User defined comments. Maximum of 1999 characters.
target_type={value}	(Optional) Specify the target type.
username={value}	(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).
password={value}	(Optional) The password of the Network SSH account to be used for authentication. Maximum 100 characters (ascii).
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*.
password2={value}	(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).
login_type={value}	(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.

Parameter	Description
vault_id={value}	(Required if login_type=vault) The ID of the vault to be used to retrieve the password for login.
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" in the API user guide.
ips={value}	(Required to create record) The IP address(es) for the 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. An IP added to the Network SSH authentication record cannot added in Unix, Cisco or Checkpoint authentication records. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.
port={value}	(Optional) The port the database name is running on.
{XML File}	(Optional) XML file where you define private-key certificates. These are defined using this DTD: <platform API server>/api/2.0/fo/auth/network_ssh/network_ssh_auth_params.dtd

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=abc&port=270,17,122&clea
rtext_password=1&target_type=A10&password2=1234"
```

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=abc&comments=new%20a
uth%20record&port=270,17,122&cleartext_password=1&target_type=A10&p2_logi
n_type=vault&p2_vault_type=Thycotic%20Secret%20Server&p2_vault_id=41014&p
2_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>
<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>
```

New DTD for Private Key Certificates

DTD: <platform API server>/api/2.0/fo/auth/network_ssh/network_ssh_auth_params.dtd

```
<!-- QUALYS NETWORK_SSH_AUTH_PARAMS DTD -->
<!ELEMENT NETWORK_SSH_AUTH_PARAMS (PRIVATE_KEY_CERTIFICATES?)>
<!ELEMENT PRIVATE_KEY_CERTIFICATES (PRIVATE_KEY_CERTIFICATE)*>
<!ELEMENT PRIVATE_KEY_CERTIFICATE (ID?, PRIVATE_KEY_INFO,
```

```
PASSPHRASE_INFO?, CERTIFICATE?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT PRIVATE_KEY_INFO (DIGITAL_VAULT|PRIVATE_KEY)>
<!ATTLIST PRIVATE_KEY_INFO
        type (basic|vault) #REQUIRED>
<!ELEMENT PASSPHRASE_INFO (PASSPHRASE|DIGITAL_VAULT)>
<!ATTLIST PASSPHRASE_INFO
        type (basic|vault) #REQUIRED>
<!ELEMENT PASSPHRASE (#PCDATA)>
<!ELEMENT CERTIFICATE (#PCDATA)>
<!ATTLIST CERTIFICATE
        type (x.509|openssh) #REQUIRED>
<!ELEMENT PRIVATE_KEY (#PCDATA)>
<!ATTLIST PRIVATE_KEY
        type (rsa|dsa|ecdsa|ed25519) #REQUIRED>
<!ELEMENT DIGITAL_VAULT (VAULT_INFO_ID?, VAULT_USERNAME?, VAULT_TYPE?,
VAULT_ID?, FOLDER?, FILE?,SECRET_NAME?, SYSTEM_NAME?, END_POINT_NAME?,
END_POINT_TYPE?, END_POINT_CONTAINER?, AUTO_DISCOVER_SYSTEM_NAME?,
SYSTEM_NAME_SINGLE_HOST?, SYSTEM_TYPE?, CUSTOM_SYSTEM_TYPE?,
ACCOUNT_NAME?, AUTHORIZATION_NAME?, TARGET_NAME?, SECRET_KV_PATH?,
SECRET_KV_NAME?, SECRET_KV_KEY?, AK_SECRET_NAME?,VAULT_DEVICE_NAME?,
VAULT_DEVICE_HOST?, VAULT_APP_NAME?, VAULT_SERVICE_TYPE?)>
<!-- VAULT_USERNAME may ONLY be used if used within PASSPHRASE_INFO or
PASSWORD_INFO -->
<!ELEMENT VAULT_INFO_ID (#PCDATA)>
<!ELEMENT VAULT_USERNAME (#PCDATA)>
<!ELEMENT VAULT_TYPE (#PCDATA)>
<!ELEMENT VAULT_ID (#PCDATA)>
<!-- CyberArk PIM Suite/ CyberArk AIM -->
<!ELEMENT FOLDER (#PCDATA)>
<!ELEMENT FILE (#PCDATA)>
<!-- -->
<!-- Thycotic Secret Server -->
<!ELEMENT SECRET_NAME (#PCDATA)>
<!-- -->
<!-- Quest Vault / BeyondTrust PBPS -->
<!ELEMENT SYSTEM_NAME (#PCDATA)>
<!-- BEyondTrust PBPS -->
<!ELEMENT ACCOUNT_NAME (#PCDATA)>

<!-- -->
<!-- CA Access Control -->
<!ELEMENT END_POINT_NAME (#PCDATA)>
<!ELEMENT END_POINT_TYPE (#PCDATA)>
<!ELEMENT END_POINT_CONTAINER (#PCDATA)>
<!-- -->
<!-- Lieberman ERPM -->
<!ELEMENT AUTO_DISCOVER_SYSTEM_NAME (#PCDATA)>
<!ELEMENT SYSTEM_NAME_SINGLE_HOST (#PCDATA)>
```

```
<!ELEMENT SYSTEM_TYPE (#PCDATA)>
<!ELEMENT CUSTOM_SYSTEM_TYPE (#PCDATA)>
<!-- WAB (WALLIX) -->
<!ELEMENT AUTHORIZATION_NAME (#PCDATA)>
<!ELEMENT TARGET_NAME (#PCDATA)>
<!-- HashiCorp -->
<!ELEMENT SECRET_KV_PATH (#PCDATA)>
<!ELEMENT SECRET_KV_NAME (#PCDATA)>
<!ELEMENT SECRET_KV_KEY (#PCDATA)>

<!-- Azure Key -->
<!ELEMENT AK_SECRET_NAME (#PCDATA)>

<!-- CA PAM -->
<!ELEMENT VAULT_DEVICE_NAME (#PCDATA)>
<!ELEMENT VAULT_DEVICE_HOST (#PCDATA)>
<!ELEMENT VAULT_APP_NAME (#PCDATA)>

<!-- ARCON PAM -->
<!ELEMENT VAULT_SERVICE_TYPE (#PCDATA)>
<!-- EOF -->
```

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&p
assword2=1234&action=update&ids=102419"
```

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>2021-04-21T06:37:07Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Updated</TEXT>
                <ID_SET>
                    <ID>102419</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```

Delete Network SSH Records

Use the following parameter to delete one or more Network SSH authentication records.

Parameter	Description
ids={value}	(Required to delete record) Network SSH auth record IDs to delete. Specify record IDs and/or ID ranges (for example, 1359-1407). Multiple entries are comma separated.

Sample - Delete Network SSH Records

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=delete&ids=4474043"
"https://qualysapi.qualys.com/api/2.0/fo/auth/network_ssh/"
```

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>
```

Enhanced Support for Database Technology Data Collection by Using Host OS Authentication Records

APIs affected	/api/2.0/fo/subscription/option_profile/pc/?action=update|create /api/2.0/fo/subscription/option_profile/pc/?action=list api/2.0/fo/subscription/option_profile/?action=export api/2.0/fo/subscription/option_profile/?action=import
New or Updated API	Updated
DTD or XSD changes	Yes

In Policy Compliance, you have an option to enable database instance data collection by using the underlying OS authentication records without creating an authentication record for the database technology. In this release, we have extended this support. We now support the OS authentication record-based data collection for the following technologies as well:

Database	Supported Versions	Which OS Record to Use?
IBM DB2	IBM DB2 9.x IBM DB2 10.x IBM DB2 11.x	UNIX (with Sudo as root delegation) OR Windows
IBM Informix	IBM Informix 11.x IBM Informix 12.x	UNIX
Pivotal Greenplum	Pivotal Greenplum 5.x Pivotal Greenplum 6.x	UNIX
PostgreSQL	PostgreSQL 9.x PostgreSQL 10.x PostgreSQL11.x PostgreSQL 12.x	UNIX
Sybase/SAP ASE	Sybase ASE 15.x SAP Adaptive Server Enterprise 16.x	UNIX

For data collection on IBM DB2 instances, you can use your UNIX (with Sudo as root delegation) or Windows authentication record depending on the host operating system. For data collection on all other database technologies listed earlier, you need a UNIX authentication record with Sudo as root delegation.

Note: If you are using database authentication records for compliance scans of these technologies, we recommend not to enable this option. Because if you enable it, you will see duplicate results in your compliance reports, one by using database authentication records and the other by using OS-based authentication records. This functionality is useful in a scenario where you have a team responsible for compliance assessment of host operating systems, which does not have access to database authentication records. In this case, if they want to scan database instances running on host assets, they can go ahead by using OS-based authentication records.

We've updated the Option Profile APIs to incorporate this change.

Create/Update Compliance Option Profile

To enable this data collection support, while creating or updating a compliance option profile, set the value of the input parameters as mentioned in the following table. Refer to the Qualys API (VM,PC) User Guide for details on all the supported input parameters.

Input Parameters

Parameter	Description
enable_instance_data_collection={**0**|1}}	(Optional) Specify 1 to enable database instance data collection by using underlying OS authentication record. By default, this option is disabled.
instance_data_collection_auth_types	(Optional) Specify the database technologies for which you want to enable OS authentication-based data collection. The valid values are: MongoDB, Oracle, MySQL, MSSQL, IBM DB2, InformixDB, Pivotal Greenplum, PostgreSQL, Sybase. You can use this parameter only if you set the value of the enable_instance_data_collection parameter to 1.

Sample create compliance option profile

In this sample, we are creating an option profile with the enable_instance_data_collection option enabled for IBM DB2, InformixDB, Pivotal Greenplum, PostgreSQL, and Sybase.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d
"action=create&title=API_Option_profile_all&scan_ports=standard&enable_in
stance_data_collection=1&instance_data_collection_auth_types=IBM DB2,
InformixDB, Pivotal Greenplum, PostgreSQL, Sybase"
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/pc/"
```

XML Ouput:

```
<?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>2021-05-04T06:26:17Z</DATETIME>
        <TEXT>Compliance Option profile successfully added.</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>4926909</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

Sample update compliance option profile

In this sample, we are updating an existing option profile to enable database instance data collection by using the underlying OS authentication record. We are enabling data collection for Sybase database instances.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d
"action=update&id=4926909&enable_instance_data_collection=1&instance_data
_collection_auth_types=Sybase"
"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>2021-05-04T06:42:39Z</DATETIME>
        <TEXT>Compliance Option profile successfully updated.</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>4926909</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

List Compliance Option Profile

In this sample, we are listing a single option profile specified by profile ID (4926909). In the XML output, you see the database technology names as the authentication types listed inside the <INSTANCE_DATA_COLLECTION> parent tag.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d
"action=list&id=4926909"
"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>4926909</ID>
            <GROUP_NAME>
              <![CDATA[Database-Instance-Data-Collection-OS-based-Auth]]>
            </GROUP_NAME>
            <GROUP_TYPE>compliance</GROUP_TYPE>
            <USER_ID>
                <![CDATA[Joe User (joe_user)]]>
            </USER_ID>
            <UNIT_ID>0</UNIT_ID>
            <SUBSCRIPTION_ID>1981985</SUBSCRIPTION_ID>
            <IS_GLOBAL>0</IS_GLOBAL>
            <UPDATE_DATE>2021-05-04T06:59:09Z</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>15</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>
```

```
<PORT_SCANNING_AND_HOST_DISCOVERY>Normal</PORT_SCANNING_AND_HOST_DISCOVER
Y>
      </PERFORMANCE>
      <DISSOLVABLE_AGENT>
        <DISSOLVABLE_AGENT_ENABLE>0</DISSOLVABLE_AGENT_ENABLE>
        <PASSWORD_AUDITING_ENABLE>
          <HAS_PASSWORD_AUDITING_ENABLE>0</HAS_PASSWORD_AUDITING_ENABLE>
        </PASSWORD_AUDITING_ENABLE>

<WINDOWS_SHARE_ENUMERATION_ENABLE>0</WINDOWS_SHARE_ENUMERATION_ENABLE>

<WINDOWS_DIRECTORY_SEARCH_ENABLE>0</WINDOWS_DIRECTORY_SEARCH_ENABLE>
      </DISSOLVABLE_AGENT>
      <FILE_INTEGRITY_MONITORING>
        <AUTO_UPDATE_EXPECTED_VALUE>0</AUTO_UPDATE_EXPECTED_VALUE>
      </FILE_INTEGRITY_MONITORING>
      <CONTROL_TYPES>
        <FIM_CONTROLS_ENABLED>0</FIM_CONTROLS_ENABLED>
        <CUSTOM_WMI_QUERY_CHECKS>0</CUSTOM_WMI_QUERY_CHECKS>
      </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_S
YN_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>
    <INSTANCE_DATA_COLLECTION>
      <DATABASES>
        <AUTHENTICATION_TYPES_LIST>
          <AUTHENTICATION_TYPE>IBM DB2</AUTHENTICATION_TYPE>
          <AUTHENTICATION_TYPE>InformixDB</AUTHENTICATION_TYPE>
          <AUTHENTICATION_TYPE>Pivotal Greenplum</AUTHENTICATION_TYPE>
```

```
            <AUTHENTICATION_TYPE>PostgreSQL</AUTHENTICATION_TYPE>
            <AUTHENTICATION_TYPE>Sybase</AUTHENTICATION_TYPE>
          </AUTHENTICATION_TYPES_LIST>
        </DATABASES>
      </INSTANCE_DATA_COLLECTION>
      <OS_BASED_INSTANCE_DISC_COLLECTION />
    </OPTION_PROFILE>
  </OPTION_PROFILES>
```

Export Compliance Option Profile

In this sample, we are exporting a single option profile specified by ID (4926909). In the XML output, you see the database technology names as the authentication types listed under inside the <INSTANCE_DATA_COLLECTION> parent tag.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X GET
"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=4926909"
```

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>4926909</ID>
            <GROUP_NAME>
               <![CDATA[Database-Instance-Data-Collection-OS-based-Auth]]>
            </GROUP_NAME>
            <GROUP_TYPE>compliance</GROUP_TYPE>
            <USER_ID>
                <![CDATA[Joe User (joe_user)]]>
            </USER_ID>
            <UNIT_ID>0</UNIT_ID>
            <SUBSCRIPTION_ID>1981985</SUBSCRIPTION_ID>
            <IS_GLOBAL>0</IS_GLOBAL>
            <UPDATE_DATE>2021-05-04T06:59:09Z</UPDATE_DATE>
        </BASIC_INFO>
        ...
        <INSTANCE_DATA_COLLECTION>
           <DATABASES>
            <AUTHENTICATION_TYPES_LIST>
              <AUTHENTICATION_TYPE>IBM DB2</AUTHENTICATION_TYPE>
               <AUTHENTICATION_TYPE>InformixDB</AUTHENTICATION_TYPE>
```

```
                <AUTHENTICATION_TYPE>Pivotal Greenplum</AUTHENTICATION_TYPE>
                 <AUTHENTICATION_TYPE>PostgreSQL</AUTHENTICATION_TYPE>
                 <AUTHENTICATION_TYPE>Sybase</AUTHENTICATION_TYPE>
              </AUTHENTICATION_TYPES_LIST>
            </DATABASES>
        </INSTANCE_DATA_COLLECTION>
    </OPTION_PROFILE>
</OPTION_PROFILES>
```

Import Compliance Option Profile

In this sample, we are importing an option profile in an input XML file to the user's account.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -H "content-type:
text/xml" -X POST --data-binary @Database_OP.xml
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/?act
ion=import"
```

Note: The Database_OP.xml file contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8" ?>
<OPTION_PROFILES>
    <OPTION_PROFILE>
        <BASIC_INFO>
            <ID>4926909</ID>
            <GROUP_NAME>
               <![CDATA[Database-Instance-Data-Collection-OS-based-Auth]]>
            </GROUP_NAME>
            <GROUP_TYPE>compliance</GROUP_TYPE>
            <USER_ID>
                <![CDATA[[Joe User (joe_user)]]>
            </USER_ID>
            <UNIT_ID>0</UNIT_ID>
            <SUBSCRIPTION_ID>1981985</SUBSCRIPTION_ID>
            <IS_GLOBAL>0</IS_GLOBAL>
            <UPDATE_DATE>2021-05-04T07:41:08Z</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>15</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>

<PORT_SCANNING_AND_HOST_DISCOVERY>Normal</PORT_SCANNING_AND_HOST_DISCOVER
Y>
            </PERFORMANCE>
            <DISSOLVABLE_AGENT>
                <DISSOLVABLE_AGENT_ENABLE>0</DISSOLVABLE_AGENT_ENABLE>
                <PASSWORD_AUDITING_ENABLE>

<HAS_PASSWORD_AUDITING_ENABLE>0</HAS_PASSWORD_AUDITING_ENABLE>
                </PASSWORD_AUDITING_ENABLE>

<WINDOWS_SHARE_ENUMERATION_ENABLE>0</WINDOWS_SHARE_ENUMERATION_ENABLE>

<WINDOWS_DIRECTORY_SEARCH_ENABLE>0</WINDOWS_DIRECTORY_SEARCH_ENABLE>
            </DISSOLVABLE_AGENT>
            <FILE_INTEGRITY_MONITORING>
               <AUTO_UPDATE_EXPECTED_VALUE>0</AUTO_UPDATE_EXPECTED_VALUE>
            </FILE_INTEGRITY_MONITORING>
            <CONTROL_TYPES>
                <FIM_CONTROLS_ENABLED>0</FIM_CONTROLS_ENABLED>
                <CUSTOM_WMI_QUERY_CHECKS>0</CUSTOM_WMI_QUERY_CHECKS>
            </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_S
YN_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>
        <INSTANCE_DATA_COLLECTION>
            <DATABASES>
             <AUTHENTICATION_TYPES_LIST>
              <AUTHENTICATION_TYPE>IBM DB2</AUTHENTICATION_TYPE>
               <AUTHENTICATION_TYPE>InformixDB</AUTHENTICATION_TYPE>
              <AUTHENTICATION_TYPE>PostgreSQL</AUTHENTICATION_TYPE>
             <AUTHENTICATION_TYPE>Pivotal Greenplum</AUTHENTICATION_TYPE>
              <AUTHENTICATION_TYPE>Sybase</AUTHENTICATION_TYPE>
             </AUTHENTICATION_TYPES_LIST>
            </DATABASES>
        </INSTANCE_DATA_COLLECTION>
        <OS_BASED_INSTANCE_DISC_COLLECTION>
            <TECHNOLOGIES>
                <TECHNOLOGY>Oracle JRE</TECHNOLOGY>
                <TECHNOLOGY>IBM WebSphere Liberty</TECHNOLOGY>
            </TECHNOLOGIES>
        </OS_BASED_INSTANCE_DISC_COLLECTION>
    </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>2021-05-04T07:48:33Z</DATETIME>
    <TEXT> Successfully imported Option profile for the subscription Id
1981985</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>108689</KEY>
        <VALUE>
                New Option Profile
        </VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Schema Update (option_profiles.xsd)

The option_profiles.xsd schema is used to validate a proper format and required elements of the option profile XML file when importing and exporting option profiles. The tags that we've added to support database technology instance data collection by using OS-based authentication records are highlighted in the following option_profiles.xsd extract.

```
<?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="CONTROL_TYPESType">
        <xs:sequence>
...

<xs:complexType name="INSTANCE_DATA_COLLECTIONType">
        <xs:sequence>
            <xs:element type="DATABASESType" name="DATABASES"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="DATABASESType">
        <xs:sequence>
            <xs:element name="AUTHENTICATION_TYPES_LIST"
type="AUTHENTICATION_TYPES_LISTType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="AUTHENTICATION_TYPES_LISTType">
        <xs:sequence>
            <xs:element name="AUTHENTICATION_TYPE" maxOccurs="unbounded">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="MongoDB"/>
                        <xs:enumeration value="Oracle"/>
                        <xs:enumeration value="MySQL"/>
                        <xs:enumeration value="MS SQL"/>
                        <xs:enumeration value="IBM DB2"/>
                        <xs:enumeration value="InformixDB"/>
                        <xs:enumeration value="Pivotal Greenplum"/>
                        <xs:enumeration value="PostgreSQL"/>
                        <xs:enumeration value="Sybase"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Filter Remediation Tickets List by Host ID

APIs affected	msp/ticket_list.php
New or Updated API	Updated
DTD or XSD changes	Yes

Users can already use ips, asset_groups, dns_contains, and network ids as input parameters to filter the list of remediation tickets for an asset. Now we're introducing a new parameter called HOST_ID that allows users to filter the list of remediation tickets for an asset based on the unique Host ID. Users cannot combine the HOST_ID parameter with the other asset parameters to filter the remediation tickets. We also introduced a new input to show the Host ID in the XML output called show_host_id. When show_host_id=1 is specified in the API request, the Host ID will appear in the output.

Input Parameters

Parameter	Description
host_id={value}	(Optional) Show remediation tickets related to a particular asset when the specific HOST_ID is provided.
show_host_id={**0**|1}	(Optional) 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.

Sample - List Remediation Tickets

In this sample, we're filtering the list of remediation tickets by Host ID and we're showing the Host ID in the XML output.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d
"host_id=355311&show_host_id=1"
"https://qualysapi.qualys.com/msp/ticket_list.php"
```

XML Ouput:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE REMEDIATION_TICKETS SYSTEM
"https://qualysapi.qualys.com/ticket_list_output.dtd">
<REMEDIATION_TICKETS>
  <HEADER>
    <USER_LOGIN>joe_user</USER_LOGIN>
    <COMPANY><![CDATA[Qualys, Inc.]]></COMPANY>
    <DATETIME>2021-05-13T04:32:07Z</DATETIME>
    <WHERE>
      <HOST_ID>355311</HOST_ID>
```

```
        <SHOW_HOST_ID>1</SHOW_HOST_ID>
      </WHERE>
    </HEADER>
    <TICKET_LIST>
      <TICKET>
        <NUMBER>404</NUMBER>
        <CREATION_DATETIME>2019-11-10T07:29:22Z</CREATION_DATETIME>
        <DUE_DATETIME>2019-11-17T07:29:22Z</DUE_DATETIME>
        <CURRENT_STATE>OPEN</CURRENT_STATE>
        <INVALID>0</INVALID>
        <ASSIGNEE>
          <NAME><![CDATA[Joe User]]></NAME>
          <EMAIL><![CDATA[joe_user@qualys.com]]></EMAIL>
          <LOGIN>joe_user</LOGIN>
        </ASSIGNEE>
       <DETECTION>
          <IP network_id="0">10.10.36.103</IP>
          <HOST_ID>355311</HOST_ID>
          <DNSNAME><![CDATA[36-103.qualys.com]]></DNSNAME>
          <SERVICE>Local</SERVICE>
       </DETECTION>
...
```

Updated DTD

DTD: <platform API server>/ticket_list_output.dtd

The elements HOST_ID and SHOW_HOST_ID were added to the DTD.

```
<!-- QUALYS TICKET LIST OUTPUT DTD -->
<!-- $Revision$ -->

<!ELEMENT REMEDIATION_TICKETS (ERROR | (HEADER,  (TICKET_LIST,
TRUNCATION?)?))>

<!-- Ticket Report error -->
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!-- Truncation warning -->
<!ELEMENT TRUNCATION (#PCDATA)>
<!ATTLIST TRUNCATION last CDATA #IMPLIED>

<!-- Information about the Ticket Report -->
<!ELEMENT HEADER (USER_LOGIN, COMPANY, DATETIME, WHERE)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT COMPANY (#PCDATA)>
<!ELEMENT DATETIME (#PCDATA)>

<!-- Search criteria -->
```

```
<!ELEMENT WHERE ((MODIFIED_SINCE_DATETIME?,UNMODIFIED_SINCE_DATETIME?,
                  TICKET_NUMBERS?, SINCE_TICKET_NUMBER?,
UNTIL_TICKET_NUMBER?,
                  STATES?, IPS?, ASSET_GROUPS?, DNS_CONTAINS?,
NETBIOS_CONTAINS?,
                  VULN_SEVERITIES?, POTENTIAL_VULN_SEVERITIES?,
                  OVERDUE?, INVALID?, TICKET_ASSIGNEE?, QIDS?,
                  SHOW_VULN_DETAILS?,
                  VULN_TITLE_CONTAINS?, VULN_DETAILS_CONTAINS?,
VENDOR_REF_CONTAINS?,NETWORK_ID?, HOST_ID?, SHOW_HOST_ID?)+) >
<!ELEMENT MODIFIED_SINCE_DATETIME (#PCDATA)>
<!ELEMENT UNMODIFIED_SINCE_DATETIME (#PCDATA)>
<!ELEMENT TICKET_NUMBERS (#PCDATA)>
<!ELEMENT SINCE_TICKET_NUMBER (#PCDATA)>
<!ELEMENT UNTIL_TICKET_NUMBER (#PCDATA)>
<!ELEMENT STATES (#PCDATA)>
<!ELEMENT IPS (#PCDATA)>
<!ELEMENT ASSET_GROUPS (#PCDATA)>
<!ELEMENT DNS_CONTAINS (#PCDATA)>
<!ELEMENT NETBIOS_CONTAINS (#PCDATA)>
<!ELEMENT VULN_SEVERITIES (#PCDATA)>
<!ELEMENT POTENTIAL_VULN_SEVERITIES (#PCDATA)>
<!ELEMENT OVERDUE (#PCDATA)>
<!ELEMENT INVALID (#PCDATA)>
<!ELEMENT TICKET_ASSIGNEE (#PCDATA)>
<!ELEMENT QIDS (#PCDATA)>
<!ELEMENT SHOW_VULN_DETAILS (#PCDATA)>
<!ELEMENT VULN_TITLE_CONTAINS (#PCDATA)>
<!ELEMENT VULN_DETAILS_CONTAINS (#PCDATA)>
<!ELEMENT VENDOR_REF_CONTAINS (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT SHOW_HOST_ID (#PCDATA)>

...

<!-- Target Asset -->
<!ELEMENT DETECTION (IP, HOST_ID?, DNSNAME?, NBHNAME?, PORT?, SERVICE?,
PROTOCOL?,
                     FQDN?, SSL?, INSTANCE?)>
<!ELEMENT IP (#PCDATA) >
<!ATTLIST IP
  network_id  CDATA  #IMPLIED
>
<!ELEMENT HOST_ID (#PCDATA)>
<!-- DNS Hostname -->
<!ELEMENT DNSNAME (#PCDATA)>
<!-- NetBios Hostname -->
<!ELEMENT NBHNAME (#PCDATA)>
...
```