Qualys Cloud Platform (VM, PC) v10.x

API Release Notes

Version 10.9

March 12, 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.

Control References Added to Compliance Posture Info API Output

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

We updated the Compliance Posture Information API output to include control references. You'll see CIS references in CIS policies, STIG references in STIG policies, and user-defined references in custom policies. This feature allows you to easily parse data based on the reference value.

Control references will appear in XML output when the API request includes details=All or details=Basic. When a control has no reference, then the reference value is empty.

The References column always appears in CSV output. When a control has no reference, then N/A appears in the References column.

Sample Stig policy in CSV Format

You'll see the new column header "Reference". In this sample, the reference value for control ID 8249 is SV-51752r1_rule.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -d
"action=list&policy_id=597463&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"
"597463","02/24/2021 09:27:22"

"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"
"7258550","10.10.10.10","Windows Server 2012 Standard 64 bit
Edition","2k12std.sample.qualys.com","2K12STD-SAMPLE","IP","8249","Status
of the 'Allow Basic authentication' setting (WinRM
client)","SERIOUS","3","53","Windows 2012
Server","Failed","Failed","02/24/2021 07:34:26","02/24/2021
07:34:26","N/A","N/A","02/24/2021 07:11:21",,"This integer value <B>X</B>
indicates the status of the setting <B>Allow Basic authentication</B>
```

```
using the registry key path
<B>HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\WinRM\Client\Al
lowBasic</B>. A value of <B>0</B> indicates the setting is
<B>Disabled</B>, A value of <B>1</B> indicates the setting is
<B>Enabled</B>.

======Expected Value(s)======

Disabled (0)

======Current Value(s) - Last updated: 02/24/2021 at 07:11:21 AM
(GMT+0000)======
Key not Found","SV-51752r1_rule"
...
```

Sample CIS Policy in XML Format

You'll see that REFERENCE appears under CONTROL_LIST in the GLOSSARY section of the XML. In this sample, the reference value for control ID 9705 is 1.1.1.1.a.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -d
"action=list&policy_id=7256700&details=All&output_format=xml"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/"
```

XML output:

```
...
    <GLOSSARY>
      <HOST_LIST>
        <HOST>
          <ID>1574730</ID>
          <IP>10.10.10.10</IP>
          <TRACKING_METHOD>IP</TRACKING_METHOD>
          <DNS><![CDATA[centos.sample.qualys.com]]></DNS>
          <DNS_DATA>
            <HOSTNAME><![CDATA[centos]]></HOSTNAME>
            <DOMAIN><![CDATA[sample.qualys.com]]></DOMAIN>
            <FQDN><![CDATA[centos.sample.qualys.com]]></FQDN>
          </DNS_DATA>
          <OS><![CDATA[CentOS Linux 6.0]]></OS>
          <LAST_COMPLIANCE_SCAN_DATETIME>2021-02-
24T05:37:44Z</LAST_COMPLIANCE_SCAN_DATETIME>
          <PERCENTAGE><![CDATA[62.50% (280 of 448)]]></PERCENTAGE>
        </HOST>
      </HOST_LIST>
      <CONTROL_LIST>
```

```
        <CONTROL>
          <ID>9705</ID>
          <STATEMENT><![CDATA[Status of the cramfs Filesystems
(modprobe)]]></STATEMENT>
          <CRITICALITY>
            <LABEL><![CDATA[MEDIUM]]></LABEL>
            <VALUE>2</VALUE>
          </CRITICALITY>
          <REFERENCE><![CDATA[1.1.1.1.a]]></REFERENCE>
          <RATIONALE_LIST>
            <RATIONALE>
              <TECHNOLOGY_ID>43</TECHNOLOGY_ID>
              <TEXT><![CDATA[The cramfs filesystem type is a compressed
read-only Linux filesystem embedded in small footprint systems. A cramfs
image can be used without having to first decompress the image. Disable
mount utility for unneeded filesystem types reduces the local attack
surface of the server. If this filesystem type is not needed, disable
it.]]></TEXT>
            </RATIONALE>
          </RATIONALE_LIST>
        </CONTROL>
...
```

Updated DTD

<base_url>/api/2.0/fo/compliance/posture/info/posture_info_list_output.dtd

The element REFERENCE was added to the CONTROL definition in the Posture Info List Output DTD.

```
...
<!ELEMENT CONTROL_LIST (CONTROL+)>
<!ELEMENT CONTROL (ID, STATEMENT, CRITICALITY?, REFERENCE?, DEPRECATED?,
RATIONALE_LIST?)>
<!ELEMENT STATEMENT (#PCDATA)>
<!ELEMENT CRITICALITY (LABEL, VALUE)>
<!ELEMENT REFERENCE (#PCDATA)>
<!ELEMENT DEPRECATED (#PCDATA)>
<!ELEMENT RATIONALE_LIST (RATIONALE*)>
<!ELEMENT RATIONALE (TECHNOLOGY_ID, TEXT)>
...
```

Change to DNS Data in XML Output for Compliance Posture Info API

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

For the Compliance Posture Info API, we changed the way DNS data is presented in the XML output. Previously, we had a single tag for <DNS> where we showed either the DNS hostname (e.g. xpsp2-64-24-84) or the DNS hostname with the FQDN (e.g. xpsp2-64-24-84.sample.qualys.com). When scanning Windows hosts with both a scanner and an agent, the DNS value could switch between hostname only (from agent) and hostname with FQDN (from scanner). Note - There is no change to CSV output at this time.

Starting in this release, we will continue to show the <DNS> tag, but we've also added <DNS_DATA> where we'll show separate values for hostname, domain and FQDN so you can more easily parse this data and avoid confusion.

See the example below:

```
<DNS>
    <![CDATA[xpsp2-64-24-84.sample.qualys.com]]>
</DNS>
<DNS_DATA>
    <HOSTNAME>
        <![CDATA[xpsp2-64-24-84]]>
    </HOSTNAME>
    <DOMAIN>
        <![CDATA[sample.qualys.com]]>
    </DOMAIN>
    <FQDN>
        <![CDATA[xpsp2-64-24-84.sample.qualys.com]]>
    </FQDN>
</DNS_DATA>
```

Posture Info API

Here's a sample API call and output for Posture Info API in XML format. The Host details in the Glossary section of the output is where you'll find <DNS_DATA>.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=list&details=All&policy_id=3433470&output_format=xml"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/"
```

XML output:

```
...
 <GLOSSARY>
      <HOST_LIST>
        <HOST>
          <ID>77951</ID>
          <IP>10.10.24.57</IP>
          <TRACKING_METHOD>IP</TRACKING_METHOD>
          <DNS><![CDATA[sybase-24-57.sample.qualys.com]]></DNS>
          <DNS_DATA>
            <HOSTNAME><![CDATA[sybase-24-57]]></HOSTNAME>
            <DOMAIN><![CDATA[sample.qualys.com]]></DOMAIN>
            <FQDN><![CDATA[sybase-24-57.sample.qualys.com]]></FQDN>
          </DNS_DATA>
          <NETBIOS><![CDATA[SYBASE-24-57]]></NETBIOS>
          <OS><![CDATA[Windows 2003 R2 Service Pack 2]]></OS>

<OS_CPE><![CDATA[cpe:/o:microsoft:windows_2003_server::r2::]]></OS_CPE>
        </HOST>
        <HOST>
...
```

DTD update:

You'll now see DNS_DATA in the DTD for the Posture Info API.

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

```
...
<!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?, 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)>
...
```

Asset Tag Support in Windows and Unix Authentication Records

APIs affected	/api/2.0/fo/auth/windows/?action=create|update
New or Updated API	Updated
DTD or XSD changes	No
APIs affected	/api/2.0/fo/auth/unix/?action=create|update
New or Updated API	Updated
DTD or XSD changes	No
APIs affected	/api/2.0/fo/auth/windows/?action=list
New or Updated API	Updated
DTD or XSD changes	Yes
APIs affected	/api/2.0/fo/auth/unix/?action=list
New or Updated API	Updated
DTD or XSD changes	Yes

We're excited to introduce asset tag support for Windows and Unix authentication records. With this support, you have the option to define target hosts in your authentication record using asset tags instead of adding IP addresses/ranges to the record. At scan time, we'll resolve the asset tags in the record to IP addresses in your account and scan them using the login credentials defined in the record.

When configuring a Windows or Unix record, you'll now choose an asset type for the record. You have these options:

- IPs/Ranges (asset_type=ips) - Use this option to add IP addresses/ranges to the record.

- Asset Tags (asset_type=asset_tags) - Use this option to add tags to the record for the assets you want included. IP addresses with the selected tags already assigned will be associated with the record.

- IP Range in Tag Rule (asset_type=ip_range_tag_rule) - Use this option to add tags that have IP address ranges defined in the tag rule. All IP addresses defined in the tag rule will be associated with the record, including IPs that don't already have the tag assigned.

Prerequisites

- Asset Tagging must be enabled for your subscription

- Tag Support for Authentication Records must be enabled for your subscription

Please reach out to your Technical Account Manager or Qualys Support to have these features enabled.

Create/Update Authentication Records with Asset Tags

New tag related input parameters are supported for Windows and Unix authentication records only. Refer to the Qualys API (VM,PC) User Guide for details on all the input parameters for authentication records. There are several additional inputs which are not listed here.

Parameter	Description
asset_type={**ips**|asset_tags|ip_range_tag_rule}	(Optional) Indicates how assets will be defined in the record. Valid values are ips (the default), asset_tags, ip_range_tag_rule. When not specified, we'll use asset_type=ips ips - Specify this value to assign IP addresses/ranges to the record. asset_tags - Specify this value to add tags to the record for the assets you want included. IP addresses with the selected tags already assigned will be associated with the record. ip_range_tag_rule - Specify this value to add tags that have IP address ranges defined in the tag rule. All IP addresses defined in the tag rule will be associated with the record, including IPs that don't already have the tag assigned.
tag_set_by={**id**|name}	(Optional when asset_type=asset_tags or ip_range_tag_rule) Specify "id" (the default) to select a tag set by providing tag IDs. Specify "name" to select a tag set by providing tag names.
tags_include={tag1,tag2...}	(Required when asset_type=asset_tags or ip_range_tag_rule) Specify a tag set to include in the record. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. To specify tag names, you must also specify tag_set_by=name.
tags_exclude={tag1,tag2,...}	(Optional when asset_type=asset_tags or ip_range_tag_rule) Specify a tag set to exclude from the record. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated. To specify tag names, you must also specify tag_set_by=name.
tag_include_selector={**any**|all}	(Optional when asset_type=asset_tags or ip_range_tag_rule) Select "any" (the default) to include hosts that match at least one of the selected tags. Select "all" to include hosts that match all of the selected tags.
tag_exclude_selector={**any**|all}	(Optional when asset_type=asset_tags or ip_range_tag_rule) Select "any" (the default) to exclude hosts that match at least one of the selected tags. Select "all" to exclude hosts that match all of the selected tags.

Parameter	Description
ips={value}	(Required to create record when asset_type=ips or asset_type is not specified) The IP address(es) the server will log into using the record's credentials. Multiple entries are comma separated. (Optional to update record when asset_type=ips) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed. This parameter and the add_ips parameter or the remove_ips parameter cannot be specified in the same request.
add_ips={value}	(Optional to update record when asset_type=ips) Add IPs and/or ranges to the IPs list for this record. Multiple IPs/ranges are comma separated. This parameter and the ips parameter cannot be specified in the same request.
remove_ips={value}	(Optional to update record when asset_type=ips) IPs to be removed from your record. You may enter a combination of IPs and ranges. Multiple entries are comma separated. This parameter and the ips parameter cannot be specified in the same request.

Sample - Create Windows Record with Tags

In this sample, a new Windows record is created with asset_type=asset_tags.

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/windows/?action=create&titl
e=windows&username=root&asset_type=asset_tags&tags_include=ag1&tag_includ
e_selector=all&tags_exclude=ag20&tag_set_by=name&tag_exclude_selector=all
"
```

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>2021-03-11T00:45:31Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Created</TEXT>
```

```
                <ID_SET>
                    <ID>204027</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```

Sample - Create Unix Record with Tags

In this sample, a new Unix record is created with asset_type=ip_range_tag_rule.

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=create&title=u
nix&username=root&asset_type=ip_range_tag_rule&tags_include=7515612&tag_i
nclude_selector=all&tags_exclude=7514462&tag_exclude_selector=all"
```

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>2021-03-08T22:00:50Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Created</TEXT>
                <ID_SET>
                    <ID>204020</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```

Sample - Update Unix Record with Tags

In this sample, a Unix record is updated to include asset tags.

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=update&ids=204
025&asset_type=asset_tags&tags_include=7511515"
```

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>2021-03-10T23:16:57Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Updated</TEXT>
                <ID_SET>
                    <ID>204025</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```

List Windows Authentication Records

When you list Windows authentication records, you'll see new tag elements in the XML output when tags are defined for the record. In this sample, the Windows record has tag type "asset_tags" and the tag "WindowsOS" is included in the record.

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/windows/?action=list&ids=20
4017"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_WINDOWS_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/windows/dtd/auth_list_outpu
t.dtd">
<AUTH_WINDOWS_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2021-03-10T23:18:54Z</DATETIME>
        <AUTH_WINDOWS_LIST>
            <AUTH_WINDOWS>
                <ID>204017</ID>
                <TITLE>
                    <![CDATA[Windows]]>
                </TITLE>
                <USERNAME>
                    <![CDATA[root]]>
                </USERNAME>
```

```
                    <NTLM_V2>1</NTLM_V2>
                    <WINDOWS_DOMAIN>
                        <![CDATA[domain]]>
                    </WINDOWS_DOMAIN>
                    <TAGS>
                        <TAG_TYPE>asset_tags</TAG_TYPE>
                        <TAGS_INCLUDE>
                            <SELECTOR>any</SELECTOR>
                            <TAG>
                                <ID>7514462</ID>
                                <NAME>WindowsOS</NAME>
                            </TAG>
                        </TAGS_INCLUDE>
                        <TAGS_EXCLUDE>
                            <SELECTOR>any</SELECTOR>
                        </TAGS_EXCLUDE>
                    </TAGS>
                    <NETWORK_ID>0</NETWORK_ID>
                    <CREATED>
                        <DATETIME>2021-03-07T23:31:31Z</DATETIME>
                        <BY>joe_user</BY>
                    </CREATED>
                    <LAST_MODIFIED>
                        <DATETIME>2021-03-07T23:35:23Z</DATETIME>
                    </LAST_MODIFIED>
                </AUTH_WINDOWS>
            </AUTH_WINDOWS_LIST>
        </RESPONSE>
    </AUTH_WINDOWS_LIST_OUTPUT>
```

DTD update:

The DTD used when you list Windows authentication records was updated to include new tag elements (in bold). Please also note that the DTD was renamed. The new DTD name is:

<platform>/api/2.0/fo/auth/windows/dtd/auth_list_output.dtd

```
<!-- QUALYS AUTH_WINDOWS_LIST_OUTPUT DTD -->
        <!-- $Revision$ -->
        <!ELEMENT AUTH_WINDOWS_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_WINDOWS_LIST|ID_SET)?,
WARNING_LIST?, GLOSSARY?)>
        <!ELEMENT AUTH_WINDOWS_LIST (AUTH_WINDOWS+)>

        <!-- If WINDOWS_DOMAIN is set, then IP_SET is optional (not
specified means service selects IPs) -->
        <!ELEMENT AUTH_WINDOWS (ID, TITLE, USERNAME, NTLM?, NTLM_V2?,
KERBEROS?, WINDOWS_DOMAIN?, WINDOWS_AD_DOMAIN?, WINDOWS_AD_TRUST?,
IP_SET?, TAGS?, LOGIN_TYPE?, DIGITAL_VAULT?, NETWORK_ID?, CREATED,
LAST_MODIFIED, COMMENTS?, USE_AGENTLESS_TRACKING?, MINIMUM_SMB_VERSION?,
REQUIRE_SMB_SIGNING?)>
        <!ELEMENT ID (#PCDATA)>
        <!ELEMENT TITLE (#PCDATA)>
        <!ELEMENT USERNAME (#PCDATA)>
        <!ELEMENT NTLM (#PCDATA)>
        <!ELEMENT NTLM_V2 (#PCDATA)>
        <!ELEMENT KERBEROS (#PCDATA)>
        <!ELEMENT WINDOWS_DOMAIN (#PCDATA)>
        <!ELEMENT WINDOWS_AD_DOMAIN (#PCDATA)>
        <!ELEMENT WINDOWS_AD_TRUST (#PCDATA)>

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

        <!ELEMENT TAGS (TAG_TYPE, TAGS_INCLUDE, TAGS_EXCLUDE?)>
        <!ELEMENT TAG_TYPE (#PCDATA)>
        <!ELEMENT TAGS_INCLUDE (SELECTOR, TAG+)>
        <!ELEMENT SELECTOR (#PCDATA)>
        <!ELEMENT TAG (ID, NAME)>
        <!ELEMENT NAME (#PCDATA)>
        <!ELEMENT TAGS_EXCLUDE (SELECTOR, TAG?)>
        ...
```

List Unix Authentication Records

When you list Unix authentication records, you'll see new tag elements in the XML output when tags are defined for the record. In this sample, the Unix record has tag type "ip_range_tag_rule". The tag "UnixOS" is included, and the tag "AG123" is excluded.

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=list&ids=20402
8"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_UNIX_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/dtd/auth_list_output.d
td">
<AUTH_UNIX_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2021-03-11T00:47:13Z</DATETIME>
        <AUTH_UNIX_LIST>
            <AUTH_UNIX>
                <ID>204028</ID>
                <TITLE>
                    <![CDATA[unix]]>
                </TITLE>
                <USERNAME>
                    <![CDATA[root]]>
                </USERNAME>
                <SKIP_PASSWORD>0</SKIP_PASSWORD>
                <CLEARTEXT_PASSWORD>0</CLEARTEXT_PASSWORD>
                <TARGET_TYPE>
                    <![CDATA[Auto (default)]]>
                </TARGET_TYPE>
                <TAGS>
                    <TAG_TYPE>ip_range_tag_rule</TAG_TYPE>
                    <TAGS_INCLUDE>
                        <SELECTOR>all</SELECTOR>
                        <TAG>
                            <ID>7515612</ID>
                            <NAME>UnixOS</NAME>
                        </TAG>
                    </TAGS_INCLUDE>
                    <TAGS_EXCLUDE>
                        <SELECTOR>all</SELECTOR>
                        <TAG>
                            <ID>7514462</ID>
                            <NAME>AG123</NAME>
                        </TAG>
                    </TAGS_EXCLUDE>
                </TAGS>
                <NETWORK_ID>0</NETWORK_ID>
                <CREATED>
                    <DATETIME>2021-03-11T00:47:03Z</DATETIME>
                    <BY>joe_user</BY>
                </CREATED>
...
```

DTD update:

The DTD used when you list Unix authentication records was updated to include new tag elements (in bold). Please also note that the DTD was renamed. The new DTD name is:

<platform>/api/2.0/fo/auth/unix/dtd/auth_list_output.dtd

```
<!-- QUALYS AUTH_UNIX_LIST_OUTPUT DTD -->
        <!-- $Revision$ -->
        <!ELEMENT AUTH_UNIX_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_UNIX_LIST|ID_SET)?,
WARNING_LIST?, GLOSSARY?)>
        <!ELEMENT AUTH_UNIX_LIST (AUTH_UNIX+)>

        <!ELEMENT AUTH_UNIX (ID, TITLE, USERNAME, SKIP_PASSWORD?,
CLEARTEXT_PASSWORD?, TARGET_TYPE?, (ROOT_TOOL?|ROOT_TOOL_INFO_LIST?),
((RSA_PRIVATE_KEY?, DSA_PRIVATE_KEY?)|PRIVATE_KEY_CERTIFICATE_LIST?),
PORT?, IP_SET?, TAGS?, LOGIN_TYPE?, DIGITAL_VAULT?, NETWORK_ID?, CREATED,
LAST_MODIFIED, COMMENTS?, USE_AGENTLESS_TRACKING?,
AGENTLESS_TRACKING_PATH?, QUALYS_SHELL?)>
        <!ELEMENT ID (#PCDATA)>
        <!ELEMENT TITLE (#PCDATA)>
        <!ELEMENT USERNAME (#PCDATA)>
        <!ELEMENT SKIP_PASSWORD (#PCDATA)>
        <!ELEMENT CLEARTEXT_PASSWORD (#PCDATA)>
        <!ELEMENT TARGET_TYPE (#PCDATA)>
        <!ELEMENT ROOT_TOOL (#PCDATA)>
        <!ELEMENT ROOT_TOOL_INFO_LIST (ROOT_TOOL_INFO)*>
        <!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 TAGS (TAG_TYPE, TAGS_INCLUDE, TAGS_EXCLUDE?)>
<!ELEMENT TAG_TYPE (#PCDATA)>
<!ELEMENT TAGS_INCLUDE (SELECTOR, TAG+)>
<!ELEMENT SELECTOR (#PCDATA)>
<!ELEMENT TAG (ID, NAME)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT TAGS_EXCLUDE (SELECTOR, TAG?)>
...
```

Configure Oracle Authentication Record for Multitenant Container Database

APIs affected	/api/2.0/fo/auth/oracle/?action=create|update
New or Updated API	Updated
DTD or XSD changes	No
APIs affected	api/2.0/fo/auth/oracle/?action=list
New or Updated API	Updated
DTD or XSD changes	Yes

When you configure an Oracle authentication record, you can now specify that the record is for a Multitenant Container Database (CDB). Specify is_cdb=1 if the database is a CDB or is_cdb=0 if the database is not a CDB. Identifying the Oracle database as CDB ensures the right compliance checks are performed for multitenant technologies. Also, when the database is a CDB, we'll auto discover all of the Pluggable Databases (PDBs) within the container environment, and scan them for compliance. This saves you from having to create separate, additional Oracle records for each PDB instance.

Create/Update Oracle Authentication Record

There is one new input parameter for Oracle authentication records to indicate whether the database is a CDB. Refer to the Qualys API (VM,PC) User Guide for details on all the input parameters for Oracle records.

Parameter	Description
is_cdb={**0**|1}	(Optional) Indicates whether the database is a Container Database (CDB). Specify 1 if the database is a CDB or 0 (the default) if the database is not a CDB. When not specified, we'll use is_cdb=0.

Sample Create Oracle Record

In this sample, we're creating a new Oracle record for a Container Database.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=create&title=OracleCDB&username=OracleUser&password=Password&ips=
10.10.36.122&sid=sid&is_cdb=1"
"https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/"
```

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>2021-03-09T23:49:43Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>970942</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Sample Update Oracle Record

In this sample, we're updating an Oracle record to change the is_cdb value.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=update&ids=971725&is_cdb=1"
"https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/"
```

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>2021-03-10T18:44:15Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>971725</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

List Oracle Authentication Records

When you list Oracle authentication records with details=All or details=Basic, you'll see the IS_CDB value in the XML output for each record.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=list&details=All&ids=971725"
"https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_ORACLE_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/auth_oracle_list_out
put.dtd">
<AUTH_ORACLE_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-03-10T18:44:17Z</DATETIME>
    <AUTH_ORACLE_LIST>
      <AUTH_ORACLE>
        <ID>971725</ID>
        <TITLE><![CDATA[OracleCDB]]></TITLE>
        <USERNAME><![CDATA[OracleUser]]></USERNAME>
        <SID><![CDATA[sid]]></SID>
        <PORT>All</PORT>
        <IP_SET>
          <IP>10.10.36.122</IP>
        </IP_SET>
        <IS_CDB>1</IS_CDB>
        <WINDOWS_OS_CHECKS>0</WINDOWS_OS_CHECKS>
        <UNIX_OPATCH_CHECKS>0</UNIX_OPATCH_CHECKS>
        <UNIX_OS_CHECKS>0</UNIX_OS_CHECKS>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2021-03-10T18:30:34Z</DATETIME>
          <BY>joe_user</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2021-03-10T18:44:15Z</DATETIME>
        </LAST_MODIFIED>
        <IS_SYSTEM_CREATED>0</IS_SYSTEM_CREATED>
        <IS_ACTIVE>1</IS_ACTIVE>
        <IS_TEMPLATE>0</IS_TEMPLATE>
      </AUTH_ORACLE>
    </AUTH_ORACLE_LIST>
    <GLOSSARY>
      <USER_LIST>
        <USER>
```

```
            <USER_LOGIN>joe_user</USER_LOGIN>
            <FIRST_NAME>Joe</FIRST_NAME>
            <LAST_NAME>User</LAST_NAME>
          </USER>
        </USER_LIST>
      </GLOSSARY>
    </RESPONSE>
</AUTH_ORACLE_LIST_OUTPUT>
```

DTD update:

You'll now see IS_CDB in the DTD for the Oracle List output.

<platform>/api/2.0/fo/auth/oracle/auth_oracle_list_output.dtd

```
<!-- QUALYS AUTH_ORACLE_LIST_OUTPUT DTD -->
<!-- $Revision$ -->
<!ELEMENT AUTH_ORACLE_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_ORACLE_LIST|ID_SET)?, WARNING_LIST?,
GLOSSARY?)>
<!ELEMENT AUTH_ORACLE_LIST (AUTH_ORACLE+)>

<!ELEMENT AUTH_ORACLE (ID, TITLE, USERNAME, (SID|SERVICENAME)?, PORT?,
IP_SET?, PC_ONLY?, IS_CDB?, WINDOWS_OS_CHECKS?, WINDOWS_OS_OPTIONS?,
UNIX_OPATCH_CHECKS?, UNIX_OS_CHECKS?, UNIX_OS_OPTIONS?, NETWORK_ID?,
CREATED, LAST_MODIFIED, IS_SYSTEM_CREATED?, IS_ACTIVE?, IS_TEMPLATE?,
TEMPLATE?, COMMENTS?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT SID (#PCDATA)>
<!ELEMENT SERVICENAME (#PCDATA)>
<!ELEMENT PORT (#PCDATA)>
<!ELEMENT PC_ONLY (#PCDATA)>

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

```
<!ELEMENT IP_RANGE (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT CREATED (DATETIME, BY?)>
<!ELEMENT BY (#PCDATA)>
<!ELEMENT LAST_MODIFIED (DATETIME)>
<!ELEMENT IS_SYSTEM_CREATED (#PCDATA)>
<!ELEMENT IS_ACTIVE (#PCDATA)>
<!ELEMENT IS_TEMPLATE (#PCDATA)>
<!ELEMENT TEMPLATE (ID, TITLE)>
<!ELEMENT COMMENTS (#PCDATA)>
<!ELEMENT IS_CDB (#PCDATA)>
...
```

Support for database technology data collection by using underlying 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

Now 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. We are supporting the following database versions in this enhancement:

Database	Supported Versions
MongoDB	MongoDB 3.x MongoDB 4.x
Oracle	Oracle 12c Oracle 18c Oracle 19c
MySQL	MySQL 5.x MySQL 8.x
MSSQL	MSSQL 2012 MSSQL 2014 MSSQL 2016 MSSQL 2017 MSSQL 2019

Note: If you are using database authentication records for compliance scans, we recommend that you do not 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 have updated the Option Profile APIs with new input parameters to incorporate this change. We have also added the following definitions to the Option Profile Info DTD (option_profile_info.dtd).

```
<!ELEMENT INSTANCE_DATA_COLLECTION (DATABASES?)>
<!ELEMENT DATABASES (AUTHENTICATION_TYPES_LIST)>
<!ELEMENT AUTHENTICATION_TYPES_LIST (AUTHENTICATION_TYPE+)>
```

Create/Update Compliance Option Profile

The following table contains a new input parameters that we've introduced in the Option Profile APIs. You need these parameters while creating or updating an option profile. 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}	(Required) 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	(Required) Specify the database technologies for which you want to enable OS authentication-based data collection. The valid values are: MongoDB, Oracle, MySQL, MSSQL. 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 Oracle, MS SQL, MongoDB and MySQL.

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=Oracle,MS
SQL,MongoDB,MySQL"
"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>2021-02-01T09:15:43Z</DATETIME>
        <TEXT>Compliance Option profile successfully added.</TEXT>
```

```
            <ITEM_LIST>
                <ITEM>
                    <KEY>ID</KEY>
                    <VALUE>108430</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 MongoDB and Oracle database instances.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d
"action=update&id=108440&enable_instance_data_collection=1&instance_data_
collection_auth_types=MongoDB,Oracle"
"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>2021-02-01T09:17:00Z</DATETIME>
        <TEXT>Compliance Option profile successfully updated.</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>108440</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. 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=108430"
"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>108430</ID>
            <GROUP_NAME>
               <![CDATA[Profile-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>218748</SUBSCRIPTION_ID>
            <IS_GLOBAL>0</IS_GLOBAL>
            <UPDATE_DATE>2021-02-01T09:16:52Z</UPDATE_DATE>
        </BASIC_INFO>
        <SCAN>
            <PORTS>
                <STANDARD_SCAN>1</STANDARD_SCAN>
            </PORTS>
            <PERFORMANCE>
                <PARALLEL_SCALING>0</PARALLEL_SCALING>
                <OVERALL_PERFORMANCE>High</OVERALL_PERFORMANCE>
                <HOSTS_TO_SCAN>
                    <EXTERNAL_SCANNERS>20</EXTERNAL_SCANNERS>
                    <SCANNER_APPLIANCES>40</SCANNER_APPLIANCES>
                </HOSTS_TO_SCAN>
                <PROCESSES_TO_RUN>
                    <TOTAL_PROCESSES>15</TOTAL_PROCESSES>
                    <HTTP_PROCESSES>15</HTTP_PROCESSES>
                </PROCESSES_TO_RUN>
                <PACKET_DELAY>Short</PACKET_DELAY>

<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>MongoDB</AUTHENTICATION_TYPE>
                    <AUTHENTICATION_TYPE>Oracle</AUTHENTICATION_TYPE>
                    <AUTHENTICATION_TYPE>MySQL</AUTHENTICATION_TYPE>
                </AUTHENTICATION_TYPES_LIST>
            </DATABASES>
        </INSTANCE_DATA_COLLECTION>
    </OPTION_PROFILE>
</OPTION_PROFILES>
```

DTD update:

The newly added definitions in the Option Profile Info DTD (option_profile_info.dtd) are highlighted in bold for your reference.

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 BASIC_INFO (ID, GROUP_NAME, GROUP_TYPE, USER_ID, UNIT_ID,
SUBSCRIPTION_ID, IS_DEFAULT?, IS_GLOBAL?, IS_OFFLINE_SYNCABLE?,
UPDATE_DATE?)>
<!ELEMENT ID (#PCDATA)>
...
<!ELEMENT IGNORE_FIREWALL_GENERATED_TCP_RST (#PCDATA)>
<!ELEMENT IGNORE_ALL_TCP_RST (#PCDATA)>
<!ELEMENT IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK (#PCDATA)>
<!ELEMENT NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY (#PCDATA)>

<!ELEMENT INSTANCE_DATA_COLLECTION (DATABASES?)>
<!ELEMENT DATABASES (AUTHENTICATION_TYPES_LIST)>
<!ELEMENT AUTHENTICATION_TYPES_LIST  (AUTHENTICATION_TYPE+)>
```

Export Compliance Option Profile:

In this sample, we are exporting a single option profile specified by ID. 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=108430"
```

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>108430</ID>
            <GROUP_NAME>
               <![CDATA[Profile-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>218748</SUBSCRIPTION_ID>
                <IS_GLOBAL>0</IS_GLOBAL>
                <UPDATE_DATE>2021-02-01T09:16:52Z</UPDATE_DATE>
            </BASIC_INFO>
            <SCAN>
                <PORTS>
                    <STANDARD_SCAN>1</STANDARD_SCAN>
                </PORTS>
                <PERFORMANCE>
                    <PARALLEL_SCALING>0</PARALLEL_SCALING>
                    <OVERALL_PERFORMANCE>High</OVERALL_PERFORMANCE>
                    <HOSTS_TO_SCAN>
                        <EXTERNAL_SCANNERS>20</EXTERNAL_SCANNERS>
                        <SCANNER_APPLIANCES>40</SCANNER_APPLIANCES>
                    </HOSTS_TO_SCAN>
                    <PROCESSES_TO_RUN>
                        <TOTAL_PROCESSES>15</TOTAL_PROCESSES>
                        <HTTP_PROCESSES>15</HTTP_PROCESSES>
                    </PROCESSES_TO_RUN>
                    <PACKET_DELAY>Short</PACKET_DELAY>

<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>MongoDB</AUTHENTICATION_TYPE>
                        <AUTHENTICATION_TYPE>Oracle</AUTHENTICATION_TYPE>
                        <AUTHENTICATION_TYPE>MySQL</AUTHENTICATION_TYPE>
                    </AUTHENTICATION_TYPES_LIST>
                </DATABASES>
            </INSTANCE_DATA_COLLECTION>
        </OPTION_PROFILE>
</OPTION_PROFILES>
```

Sample Option Profile: Import

API Request:

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

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>2021-02-22T05:48:33Z</DATETIME>
    <TEXT>Successfully imported Option profile for the subscription Id
218748</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>108689</KEY>
        <VALUE>Database_Instance_Data_Collection_OS-Auth_API</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. To support database instance data collection by using OS-based authentication records, we have added the <xs:element type="INSTANCE_DATA_COLLECTIONType" name="INSTANCE_DATA_COLLECTION"/> tag to the XSD file.

```
<?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="OPTION_PROFILEType">
        <xs:sequence>
            <xs:element type="BASIC_INFOType" name="BASIC_INFO"/>
            <xs:element type="SCANType" name="SCAN"/>
            <xs:element type="MAPType" name="MAP" minOccurs="0"/>
            <xs:element type="ADDITIONALType" name="ADDITIONAL"/>
            <xs:element type="INSTANCE_DATA_COLLECTIONType"
name="INSTANCE_DATA_COLLECTION"/>
        </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="MSSQL"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:schema>
```

Support for data collection on technology instances by using underlying 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

Now you can enable data collection on technology instances in your environment by using the underlying OS authentication records without creating an authentication record for the Oracle JRE technology. In this release, we've added support for the Oracle Java SE Runtime Environment (JRE) 8 instances.

We've updated the Option Profile APIs with new input parameters to incorporate this change. We've also added the following definitions to the Option Profile Info DTD (option_profile_info.dtd).

```
<!ELEMENT OS_BASED_INSTANCE_DISC_COLLECTION (TECHNOLOGIES?)>
<!ELEMENT TECHNOLOGIES (TECHNOLOGY)>
<!ELEMENT TECHNOLOGY (#PCDATA)>
```

Create/Update Compliance Option Profile

Use the new input parameter while creating or updating an option profile. Refer to the Qualys API (VM,PC) User Guide for details on all the supported input parameters.

Input Parameter

Parameter	Description
os_based_instance_disc_technologies	(Required) Specify a comma-separated list of technologies to enable OS authentication-based data collection. Currently we support Oracle JRE only. Hence, the valid value is: Oracle JRE.

Sample create compliance option profile

In this sample, we are creating an option profile for data collection on Oracle JRE technology instances.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d
"action=create&id=%option_profile_id%&os_based_instance_disc_technologies
=Oracle JRE"
"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>2021-03-09T15:40:06Z</DATETIME>
        <TEXT>Compliance Option profile successfully added.</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>108962</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

Sample update compliance option profile

In this sample, we are updating an existing option profile (ID 108961) to enable data collection on Oracle JRE instances by using the underlying OS authentication record.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d
"action=update&id=108961&"
"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>2021-03-09T15:39:20Z</DATETIME>
        <TEXT>Compliance Option profile successfully updated.</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>108961</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. In the XML output, you see Oracle JRE in the <TECHNOLOGIES> tag inside the <OS_BASED_INSTANCE_DISC_COLLECTION> parent tag.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X POST -d
"action=list&id=108696"
"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>108696</ID>
            <GROUP_NAME>
                <![CDATA[Profile-Tech-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>232602</SUBSCRIPTION_ID>
            <IS_GLOBAL>0</IS_GLOBAL>
            <UPDATE_DATE>2021-03-09T07:41:08Z</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>
                ...
        <INSTANCE_DATA_COLLECTION>
            <DATABASES>
                <AUTHENTICATION_TYPES_LIST>
```

```
                        <AUTHENTICATION_TYPE>MongoDB</AUTHENTICATION_TYPE>
                        <AUTHENTICATION_TYPE>Oracle</AUTHENTICATION_TYPE>
                    </AUTHENTICATION_TYPES_LIST>
                </DATABASES>
            </INSTANCE_DATA_COLLECTION>
            <OS_BASED_INSTANCE_DISC_COLLECTION>
                <TECHNOLOGIES>
                    <TECHNOLOGY>Oracle JRE</TECHNOLOGY>
                </TECHNOLOGIES>
            </OS_BASED_INSTANCE_DISC_COLLECTION>
        </OPTION_PROFILE>
    </OPTION_PROFILES>
```

DTD update:

The newly added definitions in the Option Profile Info DTD (option_profile_info.dtd) are highlighted in bold for your reference.

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,
INSTANCE_DATA_COLLECTION,OS_BASED_INSTANCE_DISC_COLLECTION)>
<!ELEMENT BASIC_INFO (ID, GROUP_NAME, GROUP_TYPE, USER_ID, UNIT_ID,
SUBSCRIPTION_ID, IS_DEFAULT?, IS_GLOBAL?, IS_OFFLINE_SYNCABLE?,
UPDATE_DATE?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT GROUP_NAME (#PCDATA)>
<!ELEMENT GROUP_TYPE (#PCDATA)>
<!ELEMENT USER_ID (#PCDATA)>
<!ELEMENT UNIT_ID (#PCDATA)>
<!ELEMENT SUBSCRIPTION_ID (#PCDATA)>
<!ELEMENT IS_DEFAULT (#PCDATA)>
<!ELEMENT IS_GLOBAL (#PCDATA)>
<!ELEMENT IS_OFFLINE_SYNCABLE (#PCDATA)>
<!ELEMENT UPDATE_DATE (#PCDATA)>
...

<!ELEMENT INSTANCE_DATA_COLLECTION (DATABASES?)>
<!ELEMENT DATABASES (AUTHENTICATION_TYPES_LIST)>
<!ELEMENT AUTHENTICATION_TYPES_LIST  (AUTHENTICATION_TYPE+)>

<!ELEMENT OS_BASED_INSTANCE_DISC_COLLECTION (TECHNOLOGIES?)>
<!ELEMENT TECHNOLOGIES (TECHNOLOGY)>
<!ELEMENT TECHNOLOGY  (#PCDATA)>
```

Export Compliance Option Profile:

In this sample, we are exporting a single option profile specified by ID. In the XML output, you see Oracle JRE in the <TECHNOLOGIES> tag inside the <OS_BASED_INSTANCE_DISC_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=108696"
```

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>108696</ID>
            <GROUP_NAME>
                <![CDATA[Profile-Tech-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>232602</SUBSCRIPTION_ID>
            <IS_GLOBAL>0</IS_GLOBAL>
            <UPDATE_DATE>2021-03-09T07:41:08Z</UPDATE_DATE>
        </BASIC_INFO>
        <SCAN>
            ...
        </INSTANCE_DATA_COLLECTION>
        <OS_BASED_INSTANCE_DISC_COLLECTION>
            <TECHNOLOGIES>
                <TECHNOLOGY>Oracle JRE</TECHNOLOGY>
            </TECHNOLOGIES>
        </OS_BASED_INSTANCE_DISC_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 @Export_OP.xml
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/?act
ion=import"
```

Note: The Export_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>108696</ID>
            <GROUP_NAME>
                <![CDATA[ORACLE JRE6]]>
            </GROUP_NAME>
            <GROUP_TYPE>compliance</GROUP_TYPE>
            <USER_ID>
                <![CDATA[[Joe User (joe_user)]]>
            </USER_ID>
            <UNIT_ID>0</UNIT_ID>
            <SUBSCRIPTION_ID>232602</SUBSCRIPTION_ID>
            <IS_GLOBAL>0</IS_GLOBAL>
            <UPDATE_DATE>2021-03-09T07:41:08Z</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>MongoDB</AUTHENTICATION_TYPE>
                    <AUTHENTICATION_TYPE>Oracle</AUTHENTICATION_TYPE>
                </AUTHENTICATION_TYPES_LIST>
```

```
            </DATABASES>
        </INSTANCE_DATA_COLLECTION>
        <OS_BASED_INSTANCE_DISC_COLLECTION>
            <TECHNOLOGIES>
                <TECHNOLOGY>Oracle JRE</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>2021-02-22T05:48:33Z</DATETIME>
    <TEXT> Successfully imported Option profile for the subscription Id
218748</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 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="OPTION_PROFILEType">
        <xs:sequence>
```

```
            <xs:element type="BASIC_INFOType" name="BASIC_INFO"/>
            <xs:element type="SCANType" name="SCAN"/>
            <xs:element type="MAPType" name="MAP" minOccurs="0"/>
            <xs:element type="ADDITIONALType" name="ADDITIONAL"/>
            <xs:element type="INSTANCE_DATA_COLLECTIONType"
name="INSTANCE_DATA_COLLECTION"/>
            <xs:element type="OS_BASED_INSTANCE_DISC_COLLECTIONType"
name="OS_BASED_INSTANCE_DISC_COLLECTION"/>
        </xs:sequence>
    </xs:complexType>
...
<xs:complexType name="OS_BASED_INSTANCE_DISC_COLLECTIONType">
        <xs:sequence>
            <xs:element type="TECHNOLOGIESSType" name="TECHNOLOGIES"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="TECHNOLOGIESSType">
        <xs:sequence>
            <xs:element name="TECHNOLOGY" maxOccurs="unbounded">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="Oracle JRE"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Support to Exclude Asset Tags from Compliance Policies

APIs affected	/api/2.0/fo/compliance/policy/?action=list
New or Updated API	Updated
DTD or XSD changes	Yes

Users already have the option to include asset tags in their compliance policies by adding tags to the policy in the Policy Editor (in the UI). Starting with this release, users will also have the option to exclude asset tags from their policies in the Policy Editor. This gives users more control over which assets will be evaluated for the policy. When listing policies using the API, you'll now see the excluded tags for each policy in the XML output.

We've also updated the policy_list_output.dtd to add the following definition:

```
<!ELEMENT TAG_SET_EXCLUDE (TAG_ID+)>
<!ELEMENT TAG_EXCLUDE_SELECTOR (#PCDATA)>

<!ELEMENT ASSET_TAG_LIST (ASSET_INCLUDE_TAG_LIST?,
ASSET_EXCLUDE_TAG_LIST? )>
<!ELEMENT ASSET_INCLUDE_TAG_LIST (TAG+)>
<!ELEMENT ASSET_EXCLUDE_TAG_LIST (TAG+)>
<!ELEMENT TAG (TAG_ID?, TAG_NAME?)>
<!ELEMENT TAG_NAME (#PCDATA)>
```

Sample - Compliance Policy List API

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -D headers.15
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?action=list&i
ds=602547" > Policy_include_exclude_tags.xml
```

XML Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_list_ou
tput.dtd">
<POLICY_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-03-10T09:35:50Z</DATETIME>
    <POLICY_LIST>
      <POLICY>
        <ID>602547</ID>
        <TITLE><![CDATA[Policy_include_exclude_tag]]></TITLE>
        <CREATED>
          <DATETIME>2021-03-02T09:38:37Z</DATETIME>
          <BY>joe_user</BY>
```

```
          </CREATED>
          <LAST_MODIFIED>
            <DATETIME>2021-03-09T12:50:13Z</DATETIME>
            <BY>joe_user</BY>
          </LAST_MODIFIED>
          <LAST_EVALUATED>
            <DATETIME>2021-03-10T07:09:00Z</DATETIME>
          </LAST_EVALUATED>
          <STATUS><![CDATA[active]]></STATUS>
          <IS_LOCKED>0</IS_LOCKED>
          <EVALUATE_NOW><![CDATA[yes]]></EVALUATE_NOW>
          <TAG_SET_INCLUDE>
            <TAG_ID>13591447</TAG_ID>
            <TAG_ID>13591220</TAG_ID>
          </TAG_SET_INCLUDE>
          <TAG_INCLUDE_SELECTOR>ANY</TAG_INCLUDE_SELECTOR>
          <TAG_SET_EXCLUDE>
            <TAG_ID>13693612</TAG_ID>
            <TAG_ID>13591221</TAG_ID>
          </TAG_SET_EXCLUDE>
          <TAG_EXCLUDE_SELECTOR>ALL</TAG_EXCLUDE_SELECTOR>
          <CONTROL_LIST>
            <CONTROL>
              <ID>1073</ID>
              <STATEMENT><![CDATA[Status of the 'Maximum Password Age'
setting (expiration) / Accounts having the 'password never expires' flag
set]]></STATEMENT>
              <CRITICALITY>
                <LABEL><![CDATA[URGENT]]></LABEL>
                <VALUE>5</VALUE>
              </CRITICALITY>
            </CONTROL>
...
    <GLOSSARY>
      <ASSET_TAG_LIST>
        <ASSET_INCLUDE_TAG_LIST>
          <TAG>
            <TAG_ID>13591447</TAG_ID>
            <TAG_NAME>win12_10.11.70.173</TAG_NAME>
          </TAG>
          <TAG>
            <TAG_ID>13591220</TAG_ID>
            <TAG_NAME>Win2k3AD_10.10.25.235</TAG_NAME>
          </TAG>
        </ASSET_INCLUDE_TAG_LIST>
        <ASSET_EXCLUDE_TAG_LIST>
          <TAG>
            <TAG_ID>13693612</TAG_ID>
            <TAG_NAME>win2k8</TAG_NAME>
```

```
            </TAG>
            <TAG>
              <TAG_ID>13591221</TAG_ID>
              <TAG_NAME>Centos6</TAG_NAME>
            </TAG>
          </ASSET_EXCLUDE_TAG_LIST>
        </ASSET_TAG_LIST>
      </GLOSSARY>
    </RESPONSE>
</POLICY_LIST_OUTPUT>
```

DTD update:

The newly added definitions in the Policy List Output DTD (policy_list_output.dtd) are highlighted in bold for your reference.

```
<!-- QUALYS POLICY_LIST_OUTPUT DTD -->
<!-- $Revision$ -->
<!ELEMENT POLICY_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, (POLICY_LIST|ID_SET)?, WARNING_LIST?,
GLOSSARY?)>
<!ELEMENT POLICY_LIST (POLICY+)>
<!ELEMENT POLICY (ID, TITLE, CREATED?, LAST_MODIFIED?, LAST_EVALUATED?,
STATUS?, IS_LOCKED?, EVALUATE_NOW?, ASSET_GROUP_IDS?, TAG_SET_INCLUDE?,
TAG_INCLUDE_SELECTOR?, TAG_SET_EXCLUDE?, TAG_EXCLUDE_SELECTOR?,
INCLUDE_AGENT_IPS?, CONTROL_LIST?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>

<!ELEMENT CREATED (DATETIME, BY)>
<!ELEMENT BY (#PCDATA)>

<!ELEMENT LAST_MODIFIED (DATETIME, BY)>

<!ELEMENT LAST_EVALUATED (DATETIME)>
```

```
<!ELEMENT STATUS (#PCDATA)>
<!ELEMENT IS_LOCKED (#PCDATA)>
<!ELEMENT EVALUATE_NOW (#PCDATA)>


<!ELEMENT ASSET_GROUP_IDS (#PCDATA)>
<!ATTLIST ASSET_GROUP_IDS has_hidden_data CDATA #IMPLIED>

<!ELEMENT TAG_SET_INCLUDE (TAG_ID+)>
<!ELEMENT TAG_ID (#PCDATA)>
<!ELEMENT TAG_INCLUDE_SELECTOR (#PCDATA)>

<!ELEMENT TAG_SET_EXCLUDE (TAG_ID+)>
<!ELEMENT TAG_EXCLUDE_SELECTOR (#PCDATA)>


<!ELEMENT INCLUDE_AGENT_IPS (#PCDATA)>

<!ELEMENT CONTROL_LIST (CONTROL+)>
<!ELEMENT CONTROL (ID, STATEMENT, CRITICALITY?, DEPRECATED?,
TECHNOLOGY_LIST?)>
<!ELEMENT STATEMENT (#PCDATA)>
<!ELEMENT CRITICALITY (LABEL, VALUE)>
<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT DEPRECATED (#PCDATA)>

<!ELEMENT TECHNOLOGY_LIST (TECHNOLOGY+)>
<!ELEMENT TECHNOLOGY (ID, NAME, RATIONALE, CUSTOMIZED)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT RATIONALE (#PCDATA)>
<!ELEMENT CUSTOMIZED (#PCDATA)>

<!ELEMENT ID_SET (ID|ID_RANGE)+>
<!ELEMENT ID_RANGE (#PCDATA)>

<!ELEMENT WARNING_LIST (WARNING+)>
<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>

<!ELEMENT GLOSSARY (ASSET_GROUP_LIST?, ASSET_TAG_LIST?, USER_LIST?)>

<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP+)>
<!ELEMENT ASSET_GROUP (ID, TITLE, NETWORK_ID?, IP_SET?)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT IP_SET (IP|IP_RANGE)+>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>
```

```
<!ELEMENT ASSET_TAG_LIST (ASSET_INCLUDE_TAG_LIST?,
ASSET_EXCLUDE_TAG_LIST?)>

<!ELEMENT ASSET_INCLUDE_TAG_LIST (TAG+)>
<!ELEMENT ASSET_EXCLUDE_TAG_LIST (TAG+)>
<!ELEMENT TAG (TAG_ID?, TAG_NAME?)>
<!ELEMENT TAG_NAME (#PCDATA)>

<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>

<!-- EOF -->
```

Improvements in Host List, Update, and Purge APIs

APIs affected	/api/2.0/fo/asset/host/?action=update
New or Updated API	New
DTD or XSD changes	New
APIs affected	/api/2.0/fo/asset/host/?action=list
New or Updated API	Updated
DTD or XSD changes	Yes
APIs affected	/api/2.0/fo/asset/host/?action=purge
New or Updated API	Updated (DTD change only)
DTD or XSD changes	Yes

With this release, we're introducing a new Host Update API, and we've made changes to the Host List and Host Purge APIs:

- The Host Update API (/api/2.0/fo/asset/host/?action=update) is new in this release. This API allows you to make changes to certain host attributes. This API is similar to Update IPs (/api/2.0/fo/asset/ip/?action=update) except that you can specify the host you want to update by the host ID and this API has more host filter options. The DTD for this new API is <platform>/api/2.0/fo/asset/host/dtd/update/output.dtd.

- The Host List API (/api/2.0/fo/asset/host/?action=list) was updated to show user defined attributes in the output. The DTD for the Host List API was updated and renamed. The DTD is now <platform>/api/2.0/fo/asset/host/dtd/list/output.dtd.

- The DTD for the Host Purge API (/api/2.0/fo/asset/host/?action=purge) was renamed to follow the new naming convention. The DTD is now <platform>/api/2.0/fo/asset/host/dtd/purge/output.dtd.

Host Update API

Here you can filter host assets based on input parameters and then you can update host attributes using new update parameters (new_tracking_method, new_owner, new_ud1, new_ud2, new_ud3, and new_comment).

Good to Know

- With host update API, you can update host attributes like tracking method (IP, DNS, NETBIOS), owner, user defined fields (ud1, ud2, ud3), and comments.

- You cannot update an IP to use tracking method EC2 or AGENT. Also, if an IP is already tracked by EC2 or AGENT, you cannot change the tracking method to something else. We will skip the tracking method update in these cases.

Identify the hosts you want to update

As part of the update request you'll need to tell us which hosts you want to update. You can do this in a number of ways. You can simply specify the host IDs, or you can specify IP addresses, asset group IDs or asset group titles. When specifying IP addresses or asset groups, there are additional optional input parameters available.

Specify hosts using one of these combinations of input parameters:

- ids (required) only

- ips (required) with any of these optional parameters: host_dns, host_netbios, network_id, network_name, tracking_method

- ag_ids (required) with or without tracking_method

- ag_titles (required) with or without tracking_method

These input parameters are described in more detail below.

Identify the changes you want to make

Use new input parameters to tell us the host attributes you want to change. New input parameters include new_tracking_method, new_owner, new_ud1, new_ud2, new_ud3, and new_comment. The new values you specify will overwrite the existing values, and your changes will apply to all hosts included in the API request.

Input Parameters

Use these input parameters when updating hosts.

Parameter	Description
General	
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.
Host Filters	
ids={value}	Show only certain host IDs/ranges. One or more host IDs/ranges may be specified. Multiple entries are comma separated. A host ID range is specified with a hyphen (for example, 190-400).Valid host IDs are required.

ips={value} -or- {POSTed CSV raw data}	The hosts within the subscription you want to update. IPs must be specified by using the "ips" parameter (using the POST method) or by uploading CSV raw data (using the POST method). To upload CSV raw data, specify --data-binary <data>. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.30.1-10.10.30.50). CIDR notation is supported.
network_id={value}	(Valid only when the Network Support feature is enabled for the user's account) Restrict the request to a certain custom network by specifying the network ID. When unspecified, we default to "0" for Global Default Network.
network_name={value}	(Valid only when the Network Support feature is enabled for the user's account) Restrict the request to a certain custom network by specifying the network name.
tracking_method={value}	Show only IP addresses/ranges which have a certain tracking method.
host_dns={value}	The DNS hostname for the IP you want to update. A single IP must be specified in the same request and the IP will only be updated if it matches the hostname specified.
host_netbios={value}	The NetBIOS hostname for the IP you want to update. A single IP must be specified in the same request and the IP will only be updated if it matches the hostname specified.
Host Changes	
new_tracking_method={value}	(Optional) Change the tracking method. Specify IP for IP address, DNS or NETBIOS. Note - You cannot change the tracking method to EC2 or AGENT. If an IP is already tracked by EC2 or AGENT, you cannot change the tracking method to something else.
new_owner={value}	(Optional) Change the owner of the host asset(s). The owner must be a Manager. Another user (Unit Manager, Scanner, Reader) can be the owner if the IP address is in the user's account.
new_ud1={value} new_ud2={value} new_ud3={value}	(Optional) Change values for user-defined fields 1, 2 and 3. You can specify a maximum of 128 characters (ascii) for each field value.
new_comment={value}	(Optional) Change the user-defined comments. Specify new comments for the host asset(s).

Sample - Update Host Attributes with Host IDs

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo2" "POST" -d
"action=update&ids=2332017&new_tracking_method=DNS&new_ud1=Loc&new_ud2=Fu
n&new_ud3=AT&new_comment=API_Comment&new_owner=akreb_nb"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/dtd/update/output.dtd
">
<HOST_UPDATE_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-03-09T10:38:17Z</DATETIME>
    <TEXT>Assets successfully updated</TEXT>
  </RESPONSE>
</HOST_UPDATE_OUTPUT>
```

Sample - Update Host Attributes with IPs

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo2" "POST" -d
"action=update&ips=10.10.32.31&new_tracking_method=DNS&new_ud1=Loc&new_ud
2=Fun&new_ud3=AT&new_comment=API_Comment&new_owner=akreb_nb"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/dtd/update/output.dtd
">
<HOST_UPDATE_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-03-09T06:03:42Z</DATETIME>
    <TEXT>Assets successfully updated</TEXT>
  </RESPONSE>
</HOST_UPDATE_OUTPUT>
```

Sample - Update Host Attributes with Asset Group IDs

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo2" "POST" -d
"action=update&ag_ids=4580719&new_tracking_method=IP&new_ud1=Loc&new_ud2=
Fun&new_ud3=AT&new_comment=API_Comment&new_owner=akreb_nb"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/dtd/update/output.dtd
">
<HOST_UPDATE_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-03-09T10:39:11Z</DATETIME>
    <TEXT>Assets successfully updated</TEXT>
  </RESPONSE>
</HOST_UPDATE_OUTPUT>
```

Sample - Update Host Attributes with Asset Group Titles

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo2" "POST" -d
"action=update&ag_titles=AG_Update&new_tracking_method=IP&new_ud1=Loc&new
_ud2=Fun&new_ud3=AT&new_comment=API_Comment&new_owner=akreb_nb"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/dtd/update/output.dtd
">
<HOST_UPDATE_OUTPUT>
  <RESPONSE>
    <DATETIME>2021-03-09T10:39:43Z</DATETIME>
    <TEXT>Assets successfully updated</TEXT>
  </RESPONSE>
</HOST_UPDATE_OUTPUT>
```

New DTD for Host Update

<platform>/api/2.0/fo/asset/host/dtd/update/output.dtd

```
<!-- QUALYS HOST_OUTPUT DTD FOR UPDATE ACTION-->
<!-- $Revision$ -->
<!ELEMENT HOST_UPDATE_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 specified, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>
<!ELEMENT RESPONSE (DATETIME, CODE?, TEXT, ITEM_LIST?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT ITEM_LIST (ITEM+)>
<!ELEMENT ITEM (KEY, VALUE*)>
<!-- EOF -->
```

Host List API

User-defined attributes will now appear in the XML output for the Host List API.

Sample - List Host Attributes

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -X "POST" -d
"action=list&details=All"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_OUTPUT SYSTEM
"http://qualysapi.qualys.com/api/2.0/fo/asset/host/dtd/output.dtd">
<HOST_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2021-02-01T08:23:10Z</DATETIME>
        <HOST_LIST>
            <HOST>
                <ID>138033</ID>
                <IP>10.115.75.11</IP>
                <TRACKING_METHOD>IP</TRACKING_METHOD>
                <NETWORK_ID>0</NETWORK_ID>
```

```
                    <DNS>
                        <![CDATA[10-115-75-11.bogus.tld]]>
                    </DNS>
                    <DNS_DATA>
                        <HOSTNAME>
                            <![CDATA[10-115-75-11]]>
                        </HOSTNAME>
                        <DOMAIN>
                            <![CDATA[bogus.tld]]>
                        </DOMAIN>
                        <FQDN>
                            <![CDATA[10-115-75-11.bogus.tld]]>
                        </FQDN>
                    </DNS_DATA>
                    <NETBIOS>
                        <![CDATA[SYS_10_115_75_11]]>
                    </NETBIOS>
                    <OS>
                        <![CDATA[Windows 2000 Server Service Pack 4]]>
                    </OS>
                    <LAST_VULN_SCAN_DATETIME>2018-01-
12T08:17:14Z</LAST_VULN_SCAN_DATETIME>
                    <LAST_VM_SCANNED_DATE>2018-01-
12T08:17:14Z</LAST_VM_SCANNED_DATE>
                    <LAST_VM_SCANNED_DURATION>139</LAST_VM_SCANNED_DURATION>
                    <LAST_VM_AUTH_SCANNED_DATE>2018-01-
12T08:17:14Z</LAST_VM_AUTH_SCANNED_DATE>

<LAST_VM_AUTH_SCANNED_DURATION>139</LAST_VM_AUTH_SCANNED_DURATION>
                    <USER_DEF>
                        <VALUE_1 ud_attr="Location">
                            <![CDATA[Test_location]]>
                        </VALUE_1>
                        <VALUE_2 ud_attr="Test attribute">
                            <![CDATA[Test_function]]>
                        </VALUE_2>
                        <VALUE_3 ud_attr="Asset Tag">
                            <![CDATA[Test_asset_tag]]>
                        </VALUE_3>
                    </USER_DEF>
                </HOST>
            </HOST_LIST>
            <GLOSSARY>
                <USER_DEF>
                    <LABEL_1>
                        <![CDATA[Location]]>
                    </LABEL_1>
                    <LABEL_2>
                        <![CDATA[Test attribute]]>
```

```
                    </LABEL_2>
                    <LABEL_3>
                        <![CDATA[Asset Tag]]>
                    </LABEL_3>
                </USER_DEF>
            </GLOSSARY>
        </RESPONSE>
    </HOST_LIST_OUTPUT>
```

Updated DTD for Host List

We added new elements to the DTD for user-defined attributes. Please also note that the DTD has been renamed. The new DTD name is:

<platform>/api/2.0/fo/asset/host/dtd/list/output.dtd

```
<!-- QUALYS HOST_OUTPUT DTD FOR LIST ACTION-->
<!-- $Revision$ -->
<!ELEMENT HOST_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, (HOST_LIST|ID_SET)?, WARNING?, GLOSSARY?)>
<!ELEMENT HOST_LIST (HOST+)>
<!ELEMENT HOST (ID, ASSET_ID?, IP?, IPV6?, TRACKING_METHOD?, NETWORK_ID?,
DNS?, DNS_DATA?, CLOUD_PROVIDER?, CLOUD_SERVICE?, CLOUD_RESOURCE_ID?,
EC2_INSTANCE_ID?, NETBIOS?, OS?, QG_HOSTID?, TAGS?, METADATA?,
CLOUD_PROVIDER_TAGS?, LAST_VULN_SCAN_DATETIME?, LAST_VM_SCANNED_DATE?,
LAST_VM_SCANNED_DURATION?,
LAST_VM_AUTH_SCANNED_DATE?, LAST_VM_AUTH_SCANNED_DURATION?,
LAST_COMPLIANCE_SCAN_DATETIME?, LAST_SCAP_SCAN_DATETIME?, OWNER?,
COMMENTS?, USER_DEF?, ASSET_GROUP_IDS?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT ASSET_ID (#PCDATA)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IPV6 (#PCDATA)>
<!ELEMENT TRACKING_METHOD (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT DNS (#PCDATA)>
```

```
<!ELEMENT DNS_DATA (HOSTNAME?, DOMAIN?, FQDN?)>
<!ELEMENT HOSTNAME (#PCDATA)>
<!ELEMENT DOMAIN (#PCDATA)>
<!ELEMENT FQDN (#PCDATA)>
<!ELEMENT EC2_INSTANCE_ID (#PCDATA)>
<!ELEMENT CLOUD_PROVIDER (#PCDATA)>
<!ELEMENT CLOUD_SERVICE (#PCDATA)>
<!ELEMENT CLOUD_RESOURCE_ID (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT OS (#PCDATA)>
<!ELEMENT QG_HOSTID (#PCDATA)>
<!ELEMENT TAGS (TAG*)>
<!ELEMENT TAG (TAG_ID?, NAME?)>
<!ELEMENT TAG_ID (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT LAST_VULN_SCAN_DATETIME (#PCDATA)>
<!ELEMENT LAST_VM_SCANNED_DATE (#PCDATA)>
<!ELEMENT LAST_VM_SCANNED_DURATION (#PCDATA)>
<!ELEMENT LAST_VM_AUTH_SCANNED_DATE (#PCDATA)>
<!ELEMENT LAST_VM_AUTH_SCANNED_DURATION (#PCDATA)>
<!ELEMENT LAST_COMPLIANCE_SCAN_DATETIME (#PCDATA)>
<!ELEMENT LAST_SCAP_SCAN_DATETIME (#PCDATA)>
<!ELEMENT OWNER (#PCDATA)>
<!ELEMENT COMMENTS (#PCDATA)>
<!ELEMENT USER_DEF (LABEL_1?, LABEL_2?, LABEL_3?, VALUE_1?, VALUE_2?,
VALUE_3?)>
<!ELEMENT LABEL_1 (#PCDATA)>
<!ELEMENT LABEL_2 (#PCDATA)>
<!ELEMENT LABEL_3 (#PCDATA)>
<!ELEMENT VALUE_1 (#PCDATA)>
<!ATTLIST VALUE_1
ud_attr CDATA #REQUIRED>
<!ELEMENT VALUE_2 (#PCDATA)>
<!ATTLIST VALUE_2
ud_attr CDATA #REQUIRED>
<!ELEMENT VALUE_3 (#PCDATA)>
<!ATTLIST VALUE_3
ud_attr CDATA #REQUIRED>
<!ELEMENT METADATA (EC2|GOOGLE|AZURE)+>
<!ELEMENT EC2 (ATTRIBUTE*)>
<!ELEMENT GOOGLE (ATTRIBUTE*)>
<!ELEMENT AZURE (ATTRIBUTE*)>
<!ELEMENT ATTRIBUTE
(NAME,LAST_STATUS,VALUE,LAST_SUCCESS_DATE?,LAST_ERROR_DATE?,LAST_ERROR?)>
<!ELEMENT LAST_STATUS (#PCDATA)>
<!ELEMENT LAST_SUCCESS_DATE (#PCDATA)>
<!ELEMENT LAST_ERROR_DATE (#PCDATA)>
<!ELEMENT LAST_ERROR (#PCDATA)>
<!ELEMENT CLOUD_PROVIDER_TAGS (CLOUD_TAG+)>
```

```
<!ELEMENT CLOUD_TAG (NAME, VALUE, LAST_SUCCESS_DATE)>
<!ELEMENT ASSET_GROUP_IDS (#PCDATA)>
<!ELEMENT ID_SET ((ID|ID_RANGE)+)>
<!ELEMENT ID_RANGE (#PCDATA)>
<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT GLOSSARY (USER_DEF?, USER_LIST?, ASSET_GROUP_LIST?)>
<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>
<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP+)>
<!ELEMENT ASSET_GROUP (ID, TITLE)>
<!ELEMENT TITLE (#PCDATA)>
<!-- EOF -->
```

Host Purge API

We renamed the DTD for the Purge Host API to follow the new naming convention. The new DTD name is:

<platform>/api/2.0/fo/asset/host/dtd/purge/output.dtd

```
<!-- QUALYS HOST_OUTPUT DTD FOR PURGE ACTION-->
<!-- $Revision$ -->
<!ELEMENT BATCH_RETURN (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 specified, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>
<!ELEMENT RESPONSE (DATETIME, BATCH_LIST?)>
<!ELEMENT BATCH_LIST (BATCH+)>
<!ELEMENT BATCH (CODE?, TEXT?, ID_SET?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT ID_SET (ID|ID_RANGE)+>
<!ELEMENT ID_RANGE (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!-- EOF -->
```

Vault Support for Oracle Authentication Record Resumed

APIs affected	/api/2.0/fo/auth/oracle
New or Updated API	Updated
DTD or XSD changes	No

We have resumed the vault support for Oracle Authentication Record which was discontinued after customers reported failure when creating Oracle authentication record using vault parameters. The issues are fixed in this release and now you can use respective vault parameters when creating Oracle Authentication records. Currently we support these ten vaults from API for retrieving passwords for Oracle database instances:

1) ARCON PAM

2) Azure Key

3) BeyondTrust PBPS

4) CA Access Control

5) CyberArk AIM

6) CyberArk PIM Suite

7) HashiCorp

8) Lieberman ERPM

9) Quest Vault

10) Thycotic Secret Server

API Sample - Create Oracle Record with Vault

In this sample, the authentication record is using a Thycotic Secret Server vault for retrieving the password.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=create&title=oracle&username=Qualys&ips=1.1.1.1&echo_request=1&po
rt=5857&login_type=vault&vault_type=Thycotic+Secret+Server&vault_id=33413
1&servicename=QWEB&secret_name=secret"
"https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/"
```

XML Output:

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

```
<BATCH_RETURN>
  <REQUEST>
    <DATETIME>2021-03-09T10:14:01Z</DATETIME>
    <USER_LOGIN>joe_user</USER_LOGIN>

<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/</RESOURCE>
    <PARAM_LIST>
      <PARAM>
        <KEY>action</KEY>
        <VALUE>create</VALUE>
      </PARAM>
      <PARAM>
        <KEY>title</KEY>
        <VALUE>oracle</VALUE>
      </PARAM>
      <PARAM>
        <KEY>username</KEY>
        <VALUE>Qualys</VALUE>
      </PARAM>
      <PARAM>
        <KEY>ips</KEY>
        <VALUE>1.1.1.1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>echo_request</KEY>
        <VALUE>1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>port</KEY>
        <VALUE>5857</VALUE>
      </PARAM>
      <PARAM>
        <KEY>login_type</KEY>
        <VALUE>vault</VALUE>
      </PARAM>
      <PARAM>
        <KEY>vault_type</KEY>
        <VALUE>Thycotic Secret Server</VALUE>
      </PARAM>
      <PARAM>
        <KEY>vault_id</KEY>
        <VALUE>334131</VALUE>
      </PARAM>
      <PARAM>
        <KEY>servicename</KEY>
        <VALUE>QWEB</VALUE>
      </PARAM>
      <PARAM>
        <KEY>secret_name</KEY>
```

```
        <VALUE>secret</VALUE>
      </PARAM>
    </PARAM_LIST>
  </REQUEST>
  <RESPONSE>
    <DATETIME>2021-03-09T10:14:01Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>970672</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

API Sample - Update Oracle Record with Vault

In this sample, the authentication record is using a Thycotic Secret Server vault for retrieving the password.

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=update&ids=970672&add_ips=1.1.1.20&comments=ip address Updated"
"https://qualysapi.qualys.com/api/2.0/fo/auth/oracle/"
```

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>2021-03-09T10:31:03Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>970672</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```