Qualys API Release Notes

Version 8.11

Qualys 8.11 includes improvements to the Qualys API, giving you more ways to integrate your programs and API calls with Qualys Vulnerability Management (VM) and Qualys Policy Compliance (PC). Looking for our API user guides? Just log in to your Qualys account and go to Help > Resources.

What's New

URL to the Qualys API Server

Qualys maintains multiple Qualys platforms. The Qualys API server URL that you should use for API requests depends on the platform where your account is located.

Account Location	API Server URL
Qualys US Platform 1	https://qualysapi.qualys.com
Qualys US Platform 2	https://qualysapi.qg2.apps.qualys.com
Qualys US Platform 3	https://qualysapi.qg3.apps.qualys.com
Qualys EU Platform 1	https://qualysapi.qualys.eu
Qualys EU Platform 2	https://qualysapi.qg2.apps.qualys.eu
Qualys India Platform 1	https://qualysapi.qg1.apps.qualys.in
Qualys Private Cloud Platform	https://qualysapi.<customer_base_url>

The Qualys API documentation and sample code use the API server URL for the Qualys US Platform 1. If your account is located on another platform, please replace this URL with the appropriate server URL for your account.

Tomcat Server Auth - Extended Support to Windows

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

We now support vulnerability and compliance scans for tomcat servers running on Windows hosts. Simply create a Tomcat Server record with details about your Apache Tomcat installation and instance. Your Tomcat Server records may include details for both Windows and Unix installations (previously supported).

This release includes the following updates:

1) For Windows, we support Apache Tomcat 7.x and 8.x. For Unix, we added support for Apache Tomcat 8.x. We also support these technologies for Unix: Apache Tomcat 6.x and 7.x, VMware vFabric tc Server 2.9.x and Pivotal tc Server 3.x.

2) We added new input parameters for creating and updating Tomcat Server authentication records for Windows, including installation_path_windows, instance_path_windows and service_name.

3) When listing Tomcat Server records the XML output will show the Windows installation path, instance path and service name when specified in the record. The Auth Tomcat List Output DTD was updated.

Input Parameters

Use these input parameters for specifying details about your Windows and/or Unix installation when creating and updating a Tomcat Server record. For a complete list of input parameters, please refer to the API V2 User Guide.

Parameter	Description
installation_path_windows={value}	(Optional for Windows; invalid for Unix) The Windows directory where the tomcat server is installed. When specified, at least one IP in the record must already exist in a Windows record.
	One of these parameters must be specified when creating a record: installation_path_windows, installation_path.
instance_path_windows={value}	(Optional for Windows; invalid for Unix) The Windows directory where the tomcat server instance is installed. Leave unspecified when the instance directory is the same as the Windows installation directory.

Parameter	Description
service_name={value}	(Optional for Windows; invalid for Unix) Applies to compliance scans. The Windows service name is required for certain controls that need values from the Windows registry. Enter the service name for the Apache Tomcat server running as a service.
installation_path={value}	(Optional for Unix; invalid for Windows) The Unix directory where the tomcat server is installed. When specified, at least one IP in the record must already exist in a Unix record. One of these parameters must be specified when creating a record: installation_path_windows, installation_path.
instance_path={value}	(Optional for Unix; invalid for Windows) The Unix directory where the tomcat server instance(s) are installed. You can specify a single tomcat instance (use with auto_discover_instances=0), or multiple instances (use with auto_discover_instances=1). Leave unspecified when the instance directory is the same as the Unix installation directory or when your targets have different types of tomcat servers.
auto_discover_instances={0|1}	(Optional for Unix; invalid for Windows) Specify auto_discover_instances=1 and we'll find all tomcat server instances for you. Applies to VMware vFabric and Pivotal when you've specified a directory with multiple instances or you did not specify an instance. When unspecified (auto_discover_instances=0), we will not auto discover instances. Applies to Apache Tomcat or when you've specified a single instance.

Examples

Create Tomcat Server Record

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=create&title=Tomcat&ips=10.113.197.166&installation_path_windows=
C:\tomcat\apache-tomcat-8.5.1&instance_path_windows=C:\tomcat\apache-
tomcat-8.5.1&service_name=Tomcat80&echo_request=1"
"https://qualysapi.qualys.com/api/2.0/fo/auth/tomcat/"
```

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>2017-09-13T07:37:40Z</DATETIME>
    <USER_LOGIN>qualys_joe</USER_LOGIN>

<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/auth/tomcat/</RESOURCE>
    <PARAM_LIST>
      <PARAM>
        <KEY>action</KEY>
        <VALUE>create</VALUE>
      </PARAM>
      <PARAM>
        <KEY>title</KEY>
        <VALUE>Tomcat</VALUE>
      </PARAM>
      <PARAM>
        <KEY>ips</KEY>
        <VALUE>10.113.197.166</VALUE>
      </PARAM>
      <PARAM>
        <KEY>installation_path_windows</KEY>
        <VALUE>C:\tomcat\apache-tomcat-8.5.1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>instance_path_windows</KEY>
        <VALUE>C:\tomcat\apache-tomcat-8.5.1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>service_name</KEY>
        <VALUE>Tomcat80</VALUE>
      </PARAM>
      <PARAM>
        <KEY>echo_request</KEY>
        <VALUE>1</VALUE>
      </PARAM>
    </PARAM_LIST>
  </REQUEST>
  <RESPONSE>
    <DATETIME>2017-09-13T07:37:40Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125742</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

List Tomcat Server Records

API request:

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

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_TOMCAT_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/tomcat/auth_tomcat_list_out
put.dtd">
<AUTH_TOMCAT_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2017-09-14T05:55:02Z</DATETIME>
        <AUTH_TOMCAT_LIST>
            <AUTH_TOMCAT>
                <ID>125742</ID>
                <TITLE>
                    <![CDATA[Tomcat851]]>
                </TITLE>
                <IP_SET>
                    <IP>10.113.197.166</IP>
                </IP_SET>
                <INSTALLATION_PATH_WINDOWS>
                    <![CDATA[C:\tomcat\apache-tomcat-8.5.1]]>
                </INSTALLATION_PATH_WINDOWS>
                <INSTANCE_PATH_WINDOWS>
                    <![CDATA[C:\tomcat\apache-tomcat-8.5.1]]>
                </INSTANCE_PATH_WINDOWS>
                <SERVICE_NAME_WINDOWS>
                    <![CDATA[tomcat12]]>
                </SERVICE_NAME_WINDOWS>
                <CREATED>
                    <DATETIME>2017-09-13T07:37:40Z</DATETIME>
                    <BY>qualys_joe</BY>
                </CREATED>
                <LAST_MODIFIED>
                    <DATETIME>2017-09-14T05:54:29Z</DATETIME>
                </LAST_MODIFIED>
            </AUTH_TOMCAT>
        </AUTH_TOMCAT_LIST>
    </RESPONSE>
</AUTH_TOMCAT_LIST_OUTPUT>
```

DTD update:

The Auth Tomcat List Output DTD (auth_tomcat_list_output.dtd) was updated to include new elements (in bold).

```
<!-- QUALYS AUTH_TOMCAT_LIST_OUTPUT DTD -->
<!ELEMENT AUTH_TOMCAT_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_TOMCAT_LIST|ID_SET)?, WARNING_LIST?,
                    GLOSSARY?)>
<!ELEMENT AUTH_TOMCAT_LIST (AUTH_TOMCAT+)>

<!ELEMENT AUTH_TOMCAT (ID, TITLE, IP_SET, INSTALLATION_PATH?,
INSTANCE_PATH?, AUTO_DISCOVER_INSTANCES?, INSTALLATION_PATH_WINDOWS?,
INSTANCE_PATH_WINDOWS?, SERVICE_NAME?, NETWORK_ID?, CREATED,
LAST_MODIFIED, COMMENTS?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT INSTALLATION_PATH (#PCDATA)>
<!ELEMENT INSTANCE_PATH (#PCDATA)>
<!ELEMENT AUTO_DISCOVER_INSTANCES (#PCDATA)>
<!ELEMENT INSTALLATION_PATH_WINDOWS (#PCDATA)>
<!ELEMENT INSTANCE_PATH_WINDOWS (#PCDATA)>
<!ELEMENT SERVICE_NAME (#PCDATA)>
...
```

New MongoDB Authentication API

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

With this release MongoDB authentication is supported for vulnerability scans and compliance scans using Qualys apps VM, PC, SCA. The MongoDB Record API (<baseurl>/api/2.0/fo/auth/mongodb/) allows you manage MongoDB records for performing authenticated scans of MongoDB instances running on Unix.

- Unix authentication is required for compliance scans using the PC app. Make sure the IP addresses you define in your MongoDB records are also defined in Unix records.

- We strongly recommend you create one or more dedicated user accounts to be used solely by the Qualys Cloud Platform to authenticate to MongoDB instances.

List all record types

Use the Authentication Record List API call (/api/2.0/fo/auth/?action=list). Syntax is described in the current Qualys API v2 User Guide > Chapter 8 > List Authentication Records.

API request:

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

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_RECORDS_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/auth_records.dtd">
<AUTH_RECORDS_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-09-12T22:41:47Z</DATETIME>
    <AUTH_RECORDS>
      <AUTH_UNIX_IDS>
        <ID_SET>
          <ID>13410</ID>
        </ID_SET>
      </AUTH_UNIX_IDS>
```

```
      <AUTH_WINDOWS_IDS>
        <ID_SET>
          <ID>89206</ID>
        </ID_SET>
      </AUTH_WINDOWS_IDS>
      <AUTH_MONGODB_IDS>
        <ID_SET>
          <ID>125693</ID>
          <ID>125696</ID>
          <ID>125708</ID>
        </ID_SET>
      </AUTH_MONGODB_IDS>
      <AUTH_PALO_ALTO_FIREWALL_IDS>
        <ID_SET>
          <ID>125684</ID>
        </ID_SET>
      </AUTH_PALO_ALTO_FIREWALL_IDS>
    </AUTH_RECORDS>
  </RESPONSE>
</AUTH_RECORDS_OUTPUT>
```

DTD:

<baseurl>/api/2.0/fo/auth/auth_records.dtd

New **AUTH_MONGODB_IDS** element identifies MongoDB 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?)>

<!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 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 MongoDB records

Use the new MongoDB Authentication Record List API call (/api/2.0/fo/auth/mongodb/?action=list). Syntax is described in the current Qualys API v2 User Guide > Chapter 8 > List Authentication Records by Type.

Example: List MongoDB records

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=list&details=All"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/" > file.xml
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_MONGODB_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/auth_mongodb_list_o
utput.dtd">
<AUTH_MONGODB_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-09-12T22:42:45Z</DATETIME>
    <AUTH_MONGODB_LIST>
      <AUTH_MONGODB>
        <ID>125693</ID>
        <TITLE><![CDATA[API-mongo-basic-login]]></TITLE>
        <USERNAME><![CDATA[mongo-admin-name]]></USERNAME>
        <DATABASE><![CDATA[db-admin-name]]></DATABASE>
        <PORT>28020</PORT>
<UNIX_CONFIGURATION_FILE><![CDATA[/opt/mongodb/updated]]></UNIX_CONFIGURA
TION_FILE>
        <IP_SET>
          <IP>10.20.32.239</IP>
        </IP_SET>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2017-09-12T20:22:09Z</DATETIME>
          <BY>acme_ab1</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2017-09-12T22:31:14Z</DATETIME>
        </LAST_MODIFIED>
        <COMMENTS><![CDATA[mongo-basic-login]]></COMMENTS>
      </AUTH_MONGODB>
      <AUTH_MONGODB>
        <ID>125696</ID>
        <TITLE><![CDATA[API-mongo-basic-login-with-ssl-
verify1_hosts]]></TITLE>
        <USERNAME><![CDATA[mongo-admin-name]]></USERNAME>
        <DATABASE><![CDATA[db-admin-name]></DATABASE>
        <PORT>27018</PORT>
<UNIX_CONFIGURATION_FILE><![CDATA[/opt/mongodb/]]></UNIX_CONFIGURATION_FI
LE>
        <SSL_VERIFY><![CDATA[1]]></SSL_VERIFY>
        <HOSTS>
<HOST><![CDATA[abc123.s2012r2.lab.acme.com]]></HOST>
<HOST><![CDATA[abc123.s2008r2.lab.acme.com]]></HOST>
        </HOSTS>
        <IP_SET>
          <IP>10.20.32.239</IP>
        </IP_SET>
```

```
          <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
          <NETWORK_ID>0</NETWORK_ID>
          <CREATED>
            <DATETIME>2017-09-12T20:38:19Z</DATETIME>
            <BY>acme_ab1</BY>
          </CREATED>
          <LAST_MODIFIED>
            <DATETIME>2017-09-12T20:38:19Z</DATETIME>
          </LAST_MODIFIED>
          <COMMENTS><![CDATA[mongo-basic-login-ssl_hosts]]></COMMENTS>
        </AUTH_MONGODB>
        <AUTH_MONGODB>
          <ID>125708</ID>
          <TITLE><![CDATA[API-mongo-vault-CA_Access]]></TITLE>
          <USERNAME><![CDATA[mongo-admin-name]]></USERNAME>
          <DATABASE><![CDATA[db-admin-name]><![CDATA[db-admin-name]></DATABASE>
          <PORT>27010</PORT>
<UNIX_CONFIGURATION_FILE><![CDATA[/opt/mongodb4.conf/]]></UNIX_CONFIGURAT
ION_FILE>
          <IP_SET>
            <IP>10.20.32.239</IP>
          </IP_SET>
          <LOGIN_TYPE><![CDATA[vault]]></LOGIN_TYPE>
          <DIGITAL_VAULT>
            <DIGITAL_VAULT_ID><![CDATA[166657]]></DIGITAL_VAULT_ID>
            <DIGITAL_VAULT_TYPE><![CDATA[CA Access
Control]]></DIGITAL_VAULT_TYPE>
            <DIGITAL_VAULT_TITLE><![CDATA[5-CA Access
Control]]></DIGITAL_VAULT_TITLE>
            <VAULT_EP_NAME><![CDATA[name]]></VAULT_EP_NAME>
            <VAULT_EP_TYPE><![CDATA[type]]></VAULT_EP_TYPE>
            <VAULT_EP_CONT><![CDATA[container]]></VAULT_EP_CONT>
          </DIGITAL_VAULT>
          <NETWORK_ID>0</NETWORK_ID>
          <CREATED>
            <DATETIME>2017-09-12T22:17:16Z</DATETIME>
            <BY>seenu_yn</BY>
          </CREATED>
          <LAST_MODIFIED>
            <DATETIME>2017-09-12T22:17:16Z</DATETIME>
          </LAST_MODIFIED>
          <COMMENTS><![CDATA[mongo-CA-Access-vault_login]]></COMMENTS>
        </AUTH_MONGODB>
      </AUTH_MONGODB_LIST>
      <GLOSSARY>
        <USER_LIST>
          <USER>
            <USER_LOGIN>acme_ab1</USER_LOGIN>
            <FIRST_NAME>Alan</FIRST_NAME>
```

```
          <LAST_NAME>Brooks</LAST_NAME>
        </USER>
      </USER_LIST>
    </GLOSSARY>
  </RESPONSE>
</AUTH_MONGODB_LIST_OUTPUT>
```

DTD:

<baseurl>/api/2.0/fo/auth/mongodb/auth_mongodb_list_output.dtd

```
<!-- QUALYS AUTH_MONGODB_LIST_OUTPUT DTD -->
<!ELEMENT AUTH_MONGODB_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)>
<!ELEMENT POST_DATA (#PCDATA)>
<!ELEMENT RESPONSE (DATETIME, (AUTH_MONGODB_LIST|ID_SET)?, WARNING_LIST?,
GLOSSARY?)>
<!ELEMENT AUTH_MONGODB_LIST (AUTH_MONGODB+)>
<!ELEMENT AUTH_MONGODB (ID, TITLE, USERNAME?, DATABASE, PORT,
UNIX_CONFIGURATION_FILE, SSL_VERIFY?, HOSTS?, IP_SET?, LOGIN_TYPE?,
DIGITAL_VAULT?, PRIVATE_KEY_CERTIFICATE_LIST?, NETWORK_ID?, CREATED,
LAST_MODIFIED, COMMENTS?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT PRIVATE_KEY_CERTIFICATE_LIST (PRIVATE_KEY_CERTIFICATE)*>

<!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 contents will never be rendered -->
<!ELEMENT PRIVATE_KEY EMPTY>
<!ELEMENT PASSPHRASE_INFO (DIGITAL_VAULT?)>
<!ATTLIST PASSPHRASE_INFO type (basic|vault) "basic">
<!-- Certificate contents will never be rendered -->
<!ELEMENT CERTIFICATE EMPTY>

<!ELEMENT PORT (#PCDATA)>
<!ELEMENT DATABASE (#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 LOGIN_TYPE (#PCDATA)>
<!ELEMENT UNIX_CONFIGURATION_FILE (#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)>
<!ELEMENT DIGITAL_VAULT (DIGITAL_VAULT_ID, DIGITAL_VAULT_TYPE,
DIGITAL_VAULT_TITLE, VAULT_FOLDER?, VAULT_FILE?, VAULT_SECRET_NAME?,
VAULT_SYSTEM_NAME?, VAULT_EP_NAME?, VAULT_EP_TYPE?, VAULT_EP_CONT?,
VAULT_ACCOUNT_NAME?)>
<!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_ACCOUNT_NAME (#PCDATA)>
<!-- EOF -->
```

Create / Update MongoDB record

Use these parameters to create or update a MongoDB record. For an update request, all parameters are optional except "ids" which is required.

Parameter	Description
action=create|update	(Required) The action for the API call, create or update.
ids={id1,id2,...}	(Required for update request, Invalid for create request) The IDs of the MongoDB records you want to update. Valid IDs are required. Multiple IDs are comma separated.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
title={value}	(Required for create request) A title for the Sybase record. The title must be unique. Maximum 255 characters (ascii).
ips={value}	(Required for a create request) A single IP, multiple IPs and/or ranges. Multiple entries are comma separated.
add_ips={value}	(Optional for an update request) A single IP, multiple IPs and/or ranges to be added to this record. Multiple entries are comma separated.
remove_ips={value}	(Optional for an update request) A single IP, multiple IPs and/or ranges to be removed from this record. Multiple entries are comma separated.
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.
comments={value}	(Optional) Specifies user defined notes about the MongoDB record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
unix_conf_file={value}	(Required for create request) The full path to the MongoDB configuration file on your Unix assets (IP addresses). The file must be in the same location on all assets for this record. Maximum 255 characters (ascii).
database_name={value}	(Required for create request) The username of the account to be used for authentication to the database. If password is specified this is the username of a MongoDB account. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).
port={value}	(Required for create request) The port where the database instance is running. Default is 27017.

Parameter	Description
ssl_verify={**0**|1}	(Optional) SSL verification is skipped by default. Set to 1 if you want to verify the server's certificate is valid and trusted.
hosts={value}	(Required if ssl_verify=1) A list of FQDNs for all host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed.
login_type={**basic**|vault|pkcert}	(Optional) The login type is basic by default. You can choose vault (for vault based authentication) or pkcert (for certificate based authentication).
username={value}	(For create request, required when login_type=basic or login_type=vault) The username of the MongoDB account to be used for authentication. Maximum 100 characters (ascii).
password={value}	(For create request, required when login_type=basic) The password of the MongoDB account to be used for authentication. Maximum 100 characters (ascii).
vault_type={value}	(For create request, required when login_type=vault) The vault type to be used for authentication. Supported vault_type values: BeyondTrust PBPS | CA Access Control | Cyber-Ark PIM Suite| Cyber-Ark AIM |Quest Vault | Thycotic Secret Server
vault_id={value}	(For create request, required when login_type=vault) The vault record ID to be used for authentication.
{vault parameters}	For create request, required when login_type=vault) Vault specific parameters required depend on the vault type you've selected. Vault parameters: BeyondTrust PBPS - system_name, account_name CA Access Control - end_point_name, end_point_type, end_point_container Cyber-Ark AIM and PIM - folder, file Quest - system_name Thycotic Secret Server - secret_name Looking for more details? See the Qualys API v2 User Guide > Chapter 8 > Vault Definition. Vault parameters for password, private key and passphrase: - Use the parameter private_key_vault_id to retrieve the private key from one of these vault types: BeyondTrust PBPS, Cyber-Ark AIM, Thycotic Secret Server - Use the parameter passphrase_vault_id to retrieve the passphrase from any supported vault types except BeyondTrust PBPS

Parameter	Description
private_key={value}	(For create request, required when login_type=pkcert) The private key to be used for authentication. (A vault specific private key may be defined for vault types BeyondTrust PBPS, Cyber-Ark AIM, Thycotic Secret Server.)
passphrase={value}	(For create request, required when login_type=pkcert and passphrase_vault_id is not specified) The private key passphrase value of an encrypted private key. Maximum 255 characters (ascii). (A vault specific passphrase may be defined for all vault types except BeyondTrust PBPS.)
certificate={value}	(For create request, optional when login_type=pkcert) The passphrase X.509 certificate content.

Example: Create MongoDB record - basic login

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl sample" -d
"action=create&title=API-mongodb-basic-login&username=mlqa&password=12345
abc&ips=10.20.32.239&comments=mongo-basic-login&unix_conf_path=/etc/mongo
d3.conf&port=28020&ssl_verify=0&database_name=admin"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/"> file.xml
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2017-09-12T22:43:27Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125709</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Example: Create MongoDB record - use SSL

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=create&title=API-mongo-basic-login-with-ssl-verify1_hosts&use
```

```
rname=mongo-admin&password=test123&ips=10.20.32.239&comments=mongo-
basic-login-ssl_hosts&unix_conf_path=/opt/mongodb/&port=27018&ssl_ver
ify=1&hosts=abc123.s2012r2.lab.acme.com],abc123.s2008r2.lab.acme.com"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/" > file.xml
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2017-09-12T22:45:06Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125710</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Example: Create MongoDB record - use Vault

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=create&title=API-mongo-vault-CA_Access&ips=10.20.32.239&comme
nts=mongo-CA-Access-vault_login&unix_conf_path=/opt/mongodb4.conf/&po
rt=27010&login_type=vault&vault_type=CA Access
Control&vault_id=166657&end_point_name=name&end_point_type=type&end_p
oint_container=container&username=mlqa"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/" > file.xml
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2017-09-12T22:46:47Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125711</ID>
        </ID_SET>
      </BATCH>
```

```
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Example: Update MongoDB record

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=update&ids=125693&title=API-mongo-basic-login-
updated&username=admin-updated-again&password=updated-password&databa
se_name=new-admin&comments=mongo-basic-login-updated&unix_conf_path=/
opt/mongodb/updated"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/" > file.xml
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2017-09-12T22:47:16Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>125693</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Delete MongoDB records

Use these parameters to delete one or more MongoDB records.

Parameter	Description
action=delete	(Required)
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Delete only MongoDB records with certain IDs and/or ID ranges. Valid IDs are required. Multiple entries are comma separated.

Example: Delete MongoDB records

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=delete&ids=125708,125709"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mongodb/" > file.xml
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2017-09-12T23:00:48Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Deleted</TEXT>
        <ID_SET>
          <ID_RANGE>125708-125709</ID_RANGE>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

New Palo Alto Firewall Authentication API

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

We now have added a new API to support Palo Alto Firewall. Using the Palo Alto Firewall API (.../api/2.0/fo/auth/palo_alto_firewall) you can perform these actions: create, update, list, delete.

List all record types

Supported parameters for Authentication Record List API call (/api/2.0/fo/auth/?action=list) are described in the current Qualys API v2 User Guide under Authentication Record List (in Chapter 8).

API request:

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

XML output:

```
<!-- 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?)>

<!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 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 Palo Alto Firewall records

Supported parameters for Palo Alto Firewall Authentication Record List API call (/api/2.0/fo/auth/palo_alto_firewall/?action=list) are described in the current Qualys API v2 User Guide under Authentication Record List by Type (in Chapter 8).

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=list"
"https://qualysapi.qualys.com/api/2.0/fo/auth/palo_alto_firewall/?action=
list&ids=125727"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_PALO_ALTO_FIREWALL_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/palo_alto_firewall/auth_pal
o_alto_firewall_list_output.dtd">
<AUTH_PALO_ALTO_FIREWALL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-09-13T06:30:32Z</DATETIME>
    <AUTH_PALO_ALTO_FIREWALL_LIST>
      <AUTH_PALO_ALTO_FIREWALL>
        <ID>125727</ID>
        <TITLE><![CDATA[palo-4]]></TITLE>
        <USERNAME><![CDATA[root]]></USERNAME>
        <SSL_VERIFY><![CDATA[1]]></SSL_VERIFY>
        <IP_SET>
          <IP>10.10.10.10</IP>
        </IP_SET>
        <LOGIN_TYPE><![CDATA[basic]]></LOGIN_TYPE>
        <CREATED>
          <DATETIME>2017-09-13T06:29:41Z</DATETIME>
          <BY>mayur_mm</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2017-09-13T06:29:41Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_PALO_ALTO_FIREWALL>
    </AUTH_PALO_ALTO_FIREWALL_LIST>
  </RESPONSE>
</AUTH_PALO_ALTO_FIREWALL_LIST_OUTPUT>
```

DTD:

<baseurl>/api/2.0/fo/auth/palo_alto_firewall/auth_palo_alto_firewall_list_output.dtd

```
<!-- QUALYS AUTH_PALO_ALTO_FIREWALL_LIST_OUTPUT DTD -->
    <!ELEMENT AUTH_PALO_ALTO_FIREWALL_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)>
    <!ELEMENT POST_DATA (#PCDATA)>
   <!ELEMENT RESPONSE (DATETIME, (AUTH_PALO_ALTO_FIREWALL_LIST|ID_SET)?,
WARNING_LIST?, GLOSSARY?)>
    <!ELEMENT AUTH_PALO_ALTO_FIREWALL_LIST (AUTH_PALO_ALTO_FIREWALL+)>
    <!ELEMENT AUTH_PALO_ALTO_FIREWALL (ID, TITLE, USERNAME?, SSL_VERIFY,
IP_SET?, LOGIN_TYPE?, DIGITAL_VAULT?, NETWORK_ID?, CREATED,
LAST_MODIFIED, COMMENTS?)>
    <!ELEMENT ID (#PCDATA)>
    <!ELEMENT TITLE (#PCDATA)>
    <!ELEMENT USERNAME (#PCDATA)>

    <!ELEMENT SSL_VERIFY (#PCDATA)>
    <!ELEMENT IP_SET (IP|IP_RANGE)+>
    <!ELEMENT IP (#PCDATA)>
    <!ELEMENT IP_RANGE (#PCDATA)>
    <!ELEMENT LOGIN_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)>
    <!ELEMENT DIGITAL_VAULT (DIGITAL_VAULT_ID, DIGITAL_VAULT_TYPE,
DIGITAL_VAULT_TITLE, VAULT_FOLDER?, VAULT_FILE?, VAULT_SECRET_NAME?,
VAULT_SYSTEM_NAME?, VAULT_ACCOUNT_NAME?)>
    <!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_ACCOUNT_NAME (#PCDATA)>
    <!-- EOF -->
```

Create / Update Palo Alto Firewall record

Use these parameters to create or update a Palo Alto Firewall record. For an update request, all parameters are optional except "ids" which is required.

Parameter	Description
action=create|update	(Required) The action for the API call, create or update.
ids={id1,id2,...}	(Required for update request, Invalid for create request) The IDs of the Palo Alto Firewall records you want to update. Valid IDs are required. Multiple IDs are comma separated.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
title={value}	(Required for create request) A title for the Palo Alto Firewall record. The title must be unique. Maximum 255 characters (ascii).
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 Palo Alto Firewall account. If login_type=vault is specified, this is the username of a vault account. Maximum 255 characters (ascii).
password={value}	(For create request, password or login_type=vault is required) The password of the Palo Alto Firewall account to be used for authentication. Maximum 100 characters (ascii).
login_type=vault	(For create request, password or login_type=vault is required) The password of the Palo Alto Firewall account to be used for authentication. Maximum 100 characters (ascii). Vault parameters: Vault parameters are required when login_type=vault is specified e.g. vault_id={value}, vault_type={value}, and vault specific settings. For details see the Qualys API v2 User Guide: Vault (in Chapter 8). Supported vault_type values: Cyber-Ark PIM Suite|Cyber-Ark AIM|Quest Vault|Thycotic Secret Server|BeyondTrust PBPS
comments={value}	(Optional) Specifies user defined notes about the Palo Alto Firewall record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.

Example: Create Palo Alto Firewall Record

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=create&title=palo-
4&ips=10.10.10.10&login_type=basic&username=root&password=123123"
"https://qualysapi.qualys.com/api/2.0/fo/auth/palo_alto_firewall/"
```

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>2017-09-13T06:29:41Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125727</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Example: Create Palo Alto Firewall Record using Cyber-Ark PIM Suite vault

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=create&title=palo-
4&ips=10.10.10.11&login_type=vault&username=root&vault_type=Cyber-Ark
AIM&vault_id=16034&file=file&folder=folder"
"https://qualysapi.qualys.com/api/2.0/fo/auth/palo_alto_firewall/"
```

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>2017-09-13T06:22:01Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>125726</ID>
```

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

Example: Update Palo Alto Firewall Record

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=update&ids=125726&title=Palo-5"
"https://qualysapi.qualys.com/api/2.0/fo/auth/palo_alto_firewall/"
```

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>2017-09-13T06:23:25Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>125726</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Delete Palo Alto Firewall records

Use these parameters to delete a Palo Alto Firewall record.

Parameter	Description
action=delete	(Required)
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Delete only Palo Alto Firewall records with certain IDs and/or ID ranges. Valid IDs are required. Multiple entries are comma separated.

Example: Delete Palo Alto Firewall Record

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=delete&ids=125753"
"https://qualysapi.qualys.com/api/2.0/fo/auth/palo_alto_firewall/"
```

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>2017-09-15T12:10:26Z</DATETIME>
   <BATCH_LIST>
     <BATCH>
       <TEXT>Successfully Deleted</TEXT>
       <ID_SET>
         <ID>125753</ID>
       </ID_SET>
     </BATCH>
   </BATCH_LIST>
 </RESPONSE>
</BATCH_RETURN>
```

Thycotic Secret Server vault supports private key retrieval

API affected	/api/2.0/fo/auth/mongodb/ /api/2.0/fo/auth/postgressql/ /api/2.0/fo/auth/unix/
New or Updated API	Updated
DTD or XSD changes	No

Thycotic Secret Server vault now supports the retrieval of the private key for authenticated scanning. Users can enable the private key retrieval option for these authentication records:

MongoDB record	private_key_vault_id={value}
PostgresSQL record	client_key_vault_id={value}
Unix record	{XML file} unix_auth_params.dtd

Example: Create MongoDB record - use Thycotic Secret Server Vault and retrieve the private key

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=create&title=Mongo&ips=10.20.32.239&comments=Mongodb_Auth&uni
x_conf_path=/etc/mongod1.conf&port=28021&login_type=pkcert&private_ke
y_vault_type=Thycotic Secret
Server&private_key_vault_id=166638&database_name=admin&ssl_verify=0&p
assphrase=3KT088brtc&certificate=-----BEGIN%20CERTIFICATE-----
%0D%0AMIIC%2FDCCAmUCCQCmk0zsovAvhzANBgkqhkiG9w0BAQUFADBsMQswCQYDVQQGE
wJV%0D%0AUzELMAkGA1UECAwCQ0ExFTATBgNVBAcMDFJlZHdvb2QgQ2l0eTEPMA0GA1UE
CgwG%0D%0AUXVhbHlzMRQwEgYDVQQLDAtFbmdpbmVlcmluZzESMBAGA1UEAwwJc2VydmV
yLWNh%0D%0AMB4XDTE3MDQyNzIyMTY0NFoXDTMxMDEwNDIyMTY0NFowgZQxCzAJBgNVBA
YTAlVT%0D%0AMQswCQYDVQQIDAJDQTEVMBMGA1UEBwwMUmVkd29vZCBDaXR5MQ8wDQYDV
QQKDAZR%0D%0AdWFseXMxGzAZBgNVBAsMEkVuZ2luZWVyaW5nX2NsaWVudDEzMDEGA1UE
AwwqbWxt%0D%0Ab25nb2RiMzRlLnMyMDEycjIubWxxYS5yZGxhYi5xdWFseXMuY29tMII
BIjANBgkq%0D%0AhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzYIzvdqaViRHFFOb1veMIU
VaeTXtZm70%0D%0AuyS66f01KgUkZbp%2FgTUuipMCp%2FKRTtDICGHgp%2B%2F4LHZb1
U2LYRK%2B6MggnWOl7AzF%0D%0AyG0U8rlY1%2F6XORZgmlgrtQdW%2FCABzKvEvWwhau
SGv3aE9zpazwsoFFZtTit8TT87%0D%0Azxu%2FgQ4zrCzJszunWD%2FHbXpt0%2FdN7jw
WMEgYIIbJA2ehwUQFoaVMdGPdb0%2FJGHda%0D%0AlKWlwSyE%2FySVCLRegGR7SW%2BJ
sWV15Nj2%2BzYq5ik5Pkrv7uWLUrg5q8be%2Bwo48vEm%0D%0Aa88VnXqfGbS3eYbQaSR
ldqI1Q6Tb4mUOVerEW6%2Flx1Cs7Be8TFC2ewIDAQABMA0G%0D%0ACSqGSIb3DQEBBQUA
A4GBAA8WcYVFNsWfULL%2F2TM6PHHHi41Jeosu9duFhnJoKUv1%0D%0A7JNR2qgT5Dt5s
```

```
m403tTZFpv7j59S2lOBOjv8RhsNyIRnCH43ByyM%2BBXkR4G0%0D%0AvUy7GE%2BphCdD
SPnS%2Fm7ANhkXRmEjEylXzI6JJwNwcUiemOc7S4TC69PBy4OOtLwQ%0D%0A-----
END%20CERTIFICATE-----&private_key_secret_name=mongo_key_ssh"
"https://qualysguard.qualys.com/api/2.0/fo/auth/mongodb/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
 <RESPONSE>
   <DATETIME>2017-10-06T22:12:37Z</DATETIME>
   <BATCH_LIST>
     <BATCH>
       <TEXT>Successfully Created</TEXT>
       <ID_SET>
         <ID>129696</ID>
       </ID_SET>
     </BATCH>
```

Scheduled Scan Improvements

API affected	/api/2.0/fo/schedule/scan/
New or Updated API	Updated
DTD or XSD changes	Yes

You now have the ability to update scheduled scans using the Scan Schedule V2 API (/api/2.0/fo/schedule/scan/). We also added new input parameters for more granular time selections for defining when to end, pause and resume a scan.

This release includes the following updates:

1) Ability to update a scan schedule using the API.

2) New input parameters when creating schedules, including end_after_mins, pause_after_mins, resume_in_hours. These are also available during update.

3) When listing schedules the XML output will show new settings. The Schedule Scan List DTD (schedule_scan_list_output.dtd) was updated.

Input Parameters

New input parameters are available when creating and updating schedules. For a complete list of input parameters, please refer to the API V2 User Guide.

Parameter	Description
end_after={value}	(Optional) End a scan after some number of hours. A valid value is from 1 to 119.
end_after_mins={value}	(Optional) End a scan after some number of minutes. A valid value is an integer from 0 to 59.
	Must be specified with end_after. For example, to end the scan after 2 hours and 30 minutes, you would specify end_after=2 and end_after_mins=30.
pause_after_hours={value}	(Optional) Pause a scan after some number of hours if the scan has not finished by then. A valid value is an integer from 1 to 119.
pause_after_mins={value}	(Optional) Pause a scan after some number of minutes if the scan has not finished by then. A valid value is an integer from 0-59.
	Must be specified with pause_after_hours. For example, to pause the scan after 2 hours and 30 minutes, you would specify pause_after_hours=2 and pause_after_mins=30.

Parameter	Description
resume_in_days={value}	(Optional) Resume a paused scan in some number of days. A valid value is an integer from 0 to 9 or Manually.
resume_in_hours={value}	(Optional) Resume a paused scan in some number of hours. A valid value is an integer from 0-23.
	Must be specified with pause_after_hours and resume_in_days. For example, to resume your scan in 5 hours, specify resume_in_days=0 and resume_in_hours=5. To resume your scan in 1 day and 12 hours, specify resume_in_days=1 and resume_in_hours=12. Note - The value you set for pause will determine the minimum value you can set for resume. For example, if you set the scan to pause after 1 hour then you can set it to resume in 2 or more hours. If you set the scan to pause between 1-2 hours (from 1hr, 1min to 1 hr, 59min) then you can set it to resume in 3 hours or more.
set_start_time={**0**|1}	(Optional for Update only) Specify set_start_time=1 to update any of the start time parameters.
	Must be specified with all start time parameters together: start_date, start_hour, start_minute, time_zone_code, observe_dst

Update Schedule

Use action=update with id={value} to tell us the schedule you want to update. Then make changes to the settings (same parameters used when creating a schedule). Please refer to the API V2 User Guide for full details.

Type	Parameter List
Request	action=update (required), id (required), echo_request
Scan Title	scan_title
Status	active=0|1
Option Profile	option_id or option_title
Scanner Appliance	iscanner_id, iscanner_name, default_scanner, scanners_in_ag, scanners_in_network, scanners_in_tagset
Processing Priority	priority
Asset IPs/Groups	ip, asset_group_ids or asset_groups, exclude_ip_per_scan
Asset Tags	target_from=tags, use_ip_nt_range_tags, tag_include_selector, tag_exclude_selector, tag_set_by, tag_set_exclude, tag_set_include

Type	Parameter List
EC2 Environment	connector_name or connector_uuid, ec2_endpoint, ec2_only_classic
Network	ip_network_id (when the Network Support feature is enabled)
Start Time	Must be specified together: set_start_time=1, start_date, start_hour, start_minute, time_zone_code, observe_dst
Recurrence	recurrence
Daily Scan	Must be specified together: occurrence=daily, frequency_days
Weekly Scan	Must be specified together: occurrence=weekly, frequency_weeks, weekdays
Monthly Scan	Must be specified together: occurrence=monthly, frequency_months, Nth day of month: day_of_month, Day in Nth week: day_of_week, week_of_month
End	end_after, end_after_mins
Pause and Resume	pause_after_hours, pause_after_mins, resume_in_days, resume_in_hours

Examples

Create Schedule

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=create&exclude_ip_per_scan=64.39.96.0-
64.39.111.255&echo_request=0&scan_title=My_Scan&ip=10.10.10.28&active=0&o
ccurrence=daily&recurrence=1&start_date=09/12/2017&start_hour=13&start_mi
nute=30&pause_after_hours=1&pause_after_mins=2&resume_in_days=4&resume_in
_hours=1&time_zone_code=RU-UD&option_title=Initial
Options&frequency_days=1&observe_dst=no"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2017-09-14T11:49:38Z</DATETIME>
    <TEXT>New scan scheduled successfully</TEXT>
```

```
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>146754</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Update Schedule

API request:

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

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2017-09-14T11:57:42Z</DATETIME>
    <TEXT>Edit scheduled Scan Completed successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>146754</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

List Schedules

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/?action=list&id=14
6752"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCHEDULE_SCAN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/schedule_scan_list
```

```
_output.dtd">
<SCHEDULE_SCAN_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-09-14T11:35:58Z</DATETIME>
    <SCHEDULE_SCAN_LIST>
      <SCAN>
        <ID>146752</ID>
        <ACTIVE>1</ACTIVE>
        <TITLE><![CDATA[PAUSE-MINUTE]]></TITLE>
        <USER_LOGIN>netwr_nd</USER_LOGIN>
        <TARGET><![CDATA[10.10.10.10, 10.10.10.28]]></TARGET>
        <ISCANNER_NAME><![CDATA[Default]]></ISCANNER_NAME>
        <ASSET_GROUP_TITLE_LIST>
          <ASSET_GROUP_TITLE><![CDATA[AG_2]]></ASSET_GROUP_TITLE>
        </ASSET_GROUP_TITLE_LIST>
        <EXCLUDE_IP_PER_SCAN>10.10.10.10</EXCLUDE_IP_PER_SCAN>
        <USER_ENTERED_IPS>
          <RANGE>
            <START>10.10.10.28</START>
            <END>10.10.10.28</END>
          </RANGE>
        </USER_ENTERED_IPS>
        <OPTION_PROFILE>
          <TITLE><![CDATA[Initial Options]]></TITLE>
          <DEFAULT_FLAG>1</DEFAULT_FLAG>
        </OPTION_PROFILE>
        <PROCESSING_PRIORITY>0 - No Priority</PROCESSING_PRIORITY>
        <SCHEDULE>
          <DAILY frequency_days="1" />
          <START_DATE_UTC>2017-09-13T19:31:00Z</START_DATE_UTC>
          <START_HOUR>1</START_HOUR>
          <START_MINUTE>1</START_MINUTE>
          <PAUSE_AFTER_HOURS>3</PAUSE_AFTER_HOURS>
          <PAUSE_AFTER_MINUTES>4</PAUSE_AFTER_MINUTES>
          <RESUME_IN_DAYS>2</RESUME_IN_DAYS>
          <RESUME_IN_HOURS>6</RESUME_IN_HOURS>
          <NEXTLAUNCH_UTC>2017-09-14T19:31:00</NEXTLAUNCH_UTC>
          <TIME_ZONE>
            <TIME_ZONE_CODE>IN</TIME_ZONE_CODE>
            <TIME_ZONE_DETAILS>(GMT+0530) India:
Asia/Calcutta</TIME_ZONE_DETAILS>
          </TIME_ZONE>
          <DST_SELECTED>0</DST_SELECTED>
          <MAX_OCCURRENCE>3</MAX_OCCURRENCE>
        </SCHEDULE>
      </SCAN>
    </SCHEDULE_SCAN_LIST>
  </RESPONSE>
</SCHEDULE_SCAN_LIST_OUTPUT>
```

DTD update:

The Schedule Scan List Output DTD (schedule_scan_list_output.dtd) was updated to include new elements (in bold).

```
...
<!ELEMENT SCHEDULE ((DAILY|WEEKLY|MONTHLY), START_DATE_UTC, START_HOUR,
START_MINUTE, END_AFTER_HOURS?, END_AFTER_MINUTES?, PAUSE_AFTER_HOURS?,
PAUSE_AFTER_MINUTES?, RESUME_IN_DAYS?, RESUME_IN_HOURS?, NEXTLAUNCH_UTC?,
TIME_ZONE, DST_SELECTED, MAX_OCCURRENCE?)>

...

<!-- start date of the task in UTC -->
<!ELEMENT START_DATE_UTC (#PCDATA)>
<!-- User Selected hour -->
<!ELEMENT START_HOUR (#PCDATA)>
<!-- User Selected Minute -->
<!ELEMENT START_MINUTE (#PCDATA)>
<!ELEMENT END_AFTER_HOURS (#PCDATA)>
<!ELEMENT END_AFTER_MINUTES (#PCDATA)>
<!ELEMENT PAUSE_AFTER_HOURS (#PCDATA)>
<!ELEMENT PAUSE_AFTER_MINUTES (#PCDATA)>
<!ELEMENT RESUME_IN_DAYS (#PCDATA)>
<!ELEMENT RESUME_IN_HOURS (#PCDATA)>
<!ELEMENT NEXTLAUNCH_UTC (#PCDATA)>
...
```

Scanner API - New parameter for Scanner Type

API affected	/api/2.0/fo/appliance/
New or Updated API	Updated
DTD or XSD changes	Yes

We now added a new parameter to Scanner appliance API (... /api/2.0/fo/appliance/) for you to identify the type of scanner appliance. However, the type of scanner appliance is reflected in the output only if the output mode is set to full.

Parameter	Description
action=list	(Required) A flag used to make a request for a list of scanner appliances. The GET or POST method may be used for a list request.
type={physical | virtual | offline}	(Optional) Type of scanner appliances:physical, virtual, offline

Examples

List Scanner Appliance

API request:

```
curl -u "USER:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "content-
type: application/x-www-form-urlencoded"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=list&output_mo
de=full"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE APPLIANCE_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/appliance/appliance_list_output.
dtd">
<APPLIANCE_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2017-08-31T09:14:49Z</DATETIME>
        <APPLIANCE_LIST>
            <APPLIANCE>
                <ID>132455</ID>
                <UUID>6ae4efce-0c5e-e227-82e0-1b7f55f1b98b</UUID>
                <NAME>VS_ND_1</NAME>
                <SOFTWARE_VERSION>2.6</SOFTWARE_VERSION>
                <RUNNING_SLICES_COUNT>0</RUNNING_SLICES_COUNT>
                <RUNNING_SCAN_COUNT>0</RUNNING_SCAN_COUNT>
```

```
                    <STATUS>Offline</STATUS>
                    <MODEL_NUMBER>cvscanner</MODEL_NUMBER>
                    <TYPE>Virtual</TYPE>
                    <SERIAL_NUMBER>0</SERIAL_NUMBER>
                    <ACTIVATION_CODE>15440265032293</ACTIVATION_CODE>
                    <INTERFACE_SETTINGS>
                        <INTERFACE>lan</INTERFACE>
                        <IP_ADDRESS>1.1.1.1</IP_ADDRESS>
                        <NETMASK>128.0.0.0</NETMASK>
                        <GATEWAY>128.0.0.0</GATEWAY>
                        <LEASE>Static</LEASE>
                        <IPV6_ADDRESS></IPV6_ADDRESS>
                        <SPEED></SPEED>
                        <DUPLEX>Unknown</DUPLEX>
                        <DNS>
                            <DOMAIN></DOMAIN>
                            <PRIMARY>128.0.0.0</PRIMARY>
                            <SECONDARY>128.0.0.0</SECONDARY>
                        </DNS>
                    </INTERFACE_SETTINGS>
...
          </APPLIANCE>
        </APPLIANCE_LIST>
    </RESPONSE>
</APPLIANCE_LIST_OUTPUT>
```

DTD update:

```
<!-- QUALYS APPLIANCE_LIST_OUTPUT DTD -->
<!ELEMENT APPLIANCE_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, APPLIANCE_LIST?, LICENSE_INFO?)>
        <!ELEMENT APPLIANCE_LIST (APPLIANCE+)>
            <!ELEMENT APPLIANCE (ID, UUID, NAME, NETWORK_ID?,
SOFTWARE_VERSION, RUNNING_SLICES_COUNT, RUNNING_SCAN_COUNT, STATUS,
CMD_ONLY_START?, MODEL_NUMBER?, TYPE?, SERIAL_NUMBER?, ACTIVATION_CODE?,
INTERFACE_SETTINGS*, PROXY_SETTINGS?, IS_CLOUD_DEPLOYED?, CLOUD_INFO?,
```

```
VLANS?, STATIC_ROUTES?, ML_LATEST?, ML_VERSION?, VULNSIGS_LATEST?,
VULNSIGS_VERSION?, ASSET_GROUP_COUNT?, ASSET_GROUP_LIST?,
ASSET_TAGS_LIST?, LAST_UPDATED_DATE?, POLLING_INTERVAL?, USER_LOGIN?,
HEARTBEATS_MISSED?, SS_CONNECTION?, SS_LAST_CONNECTED?, FDCC_ENABLED?,
USER_LIST?, UPDATED?, COMMENTS?, RUNNING_SCANS?, MAX_CAPACITY_UNITS?)>

...
        <!ELEMENT LICENSE_INFO (QVSA_LICENSES_COUNT, QVSA_LICENSES_USED)>
            <!ELEMENT QVSA_LICENSES_COUNT (#PCDATA)>
            <!ELEMENT QVSA_LICENSES_USED (#PCDATA)>
<!-- EOF -->
```

Option Profile API - Enable Auto Update

API affected	/api/2.0/fo/subscription/option_profile/
New or Updated API	Updated
DTD or XSD changes	Yes

We now added a new element to compliance option profile API (.../api/2.0/fo/subscription/option_profile/) when you export/import an option profile we'll now show you whether the Auto Update expected value is enabled or not.

Parameter	Description
action=export OR import	(Required) Export: The GET or POST method may be used. Import: The POST method must be used.
AUTO_UPDATE_EXPECTED_VALUE={0|1}	(Optional) Specify 1 if you want to enable the option. When you export an option profile, the value of this element indicates if the auto update option is enabled or disabled.

Examples

Export Option Profile

API request:

```
curl -u "USER:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "content-
type: application/x-www-form-urlencoded"
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/?act
ion=export&option_profile_id=137492"
```

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>137492</ID>
            <GROUP_NAME>
                <![CDATA[Windows DIC Policy OP]]>
            </GROUP_NAME>
            <GROUP_TYPE>compliance</GROUP_TYPE>
            <USER_ID>
                <![CDATA[user_john]]>
            </USER_ID>
```

```
                <UNIT_ID>0</UNIT_ID>
                <SUBSCRIPTION_ID>60</SUBSCRIPTION_ID>
                <IS_GLOBAL>1</IS_GLOBAL>
                <UPDATE_DATE>2017-10-04T17:54:03Z</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>
                <SCAN_RESTRICTION>
                    <SCAN_BY_POLICY>
                        <POLICY>
                            <ID>160219</ID>
                            <TITLE>
                                <![CDATA[Windows DIC Policy]]>
                            </TITLE>
                        </POLICY>
                    </SCAN_BY_POLICY>
                </SCAN_RESTRICTION>
                <FILE_INTEGRITY_MONITORING>
                   <AUTO_UPDATE_EXPECTED_VALUE>1</AUTO_UPDATE_EXPECTED_VALUE>
                </FILE_INTEGRITY_MONITORING>
            </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>
    </OPTION_PROFILE>
</OPTION_PROFILES>
```

Import Option Profile

To toggle the auto update expected value option via API, you can import the option profile XML by including the following element info. Set the value 1 to enable the option.

```
<AUTO_UPDATE_EXPECTED_VALUE>1</AUTO_UPDATE_EXPECTED_VALUE>
```

API request:

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

Note: “Export_OP.xml” contains the request POST data.

Request POST data (Export_OP.xml):

```
<?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>137492</ID>
            <GROUP_NAME>
                <![CDATA[Windows DIC Policy OP]]>
            </GROUP_NAME>
            <GROUP_TYPE>compliance</GROUP_TYPE>
            <USER_ID>
                <![CDATA[user_john]]>
            </USER_ID>
            <UNIT_ID>0</UNIT_ID>
            <SUBSCRIPTION_ID>60</SUBSCRIPTION_ID>
            <IS_GLOBAL>1</IS_GLOBAL>
            <UPDATE_DATE>2017-10-04T17:54:03Z</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>Norm
al</PORT_SCANNING_AND_HOST_DISCOVERY>
                </PERFORMANCE>
                <SCAN_RESTRICTION>
                    <SCAN_BY_POLICY>
                        <POLICY>
                            <ID>160219</ID>
                            <TITLE>
                                <![CDATA[Windows DIC Policy]]>
                            </TITLE>
                        </POLICY>
                    </SCAN_BY_POLICY>
                </SCAN_RESTRICTION>
                <FILE_INTEGRITY_MONITORING>
                   <AUTO_UPDATE_EXPECTED_VALUE>0</AUTO_UPDATE_EXPECTED_VALUE>
                </FILE_INTEGRITY_MONITORING>
            </SCAN>
            <ADDITIONAL>
                <HOST_DISCOVERY>
                    <TCP_PORTS>
                        <STANDARD_SCAN>1</STANDARD_SCAN>
                    </TCP_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>
        </OPTION_PROFILE>
```

```
</OPTION_PROFILES>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2017-10-05T05:00:50Z</DATETIME>
       <TEXT>Successfully imported Option profile for the subscription Id
60</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>137492</KEY>
                <VALUE>
                        Windows DIC Policy OP
                </VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

DTD update:

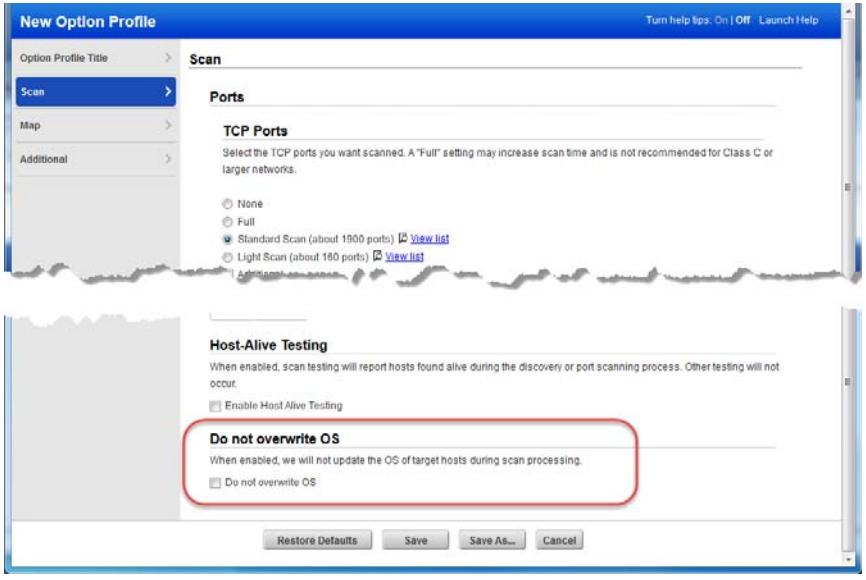
```
<!ELEMENT OPTION_PROFILES (OPTION_PROFILE)*>
<!ELEMENT OPTION_PROFILE (BASIC_INFO, SCAN, MAP?, ADDITIONAL)>
...
<!ELEMENT SCAN_BY_POLICY  (POLICY+)>
<!ELEMENT POLICY (ID, TITLE)>

<!ELEMENT FILE_INTEGRITY_MONITORING (AUTO_UPDATE_EXPECTED_VALUE?)>
<!ELEMENT AUTO_UPDATE_EXPECTED_VALUE (#PCDATA)>

<!ELEMENT CONTROL_TYPES (FIM_CONTROLS_ENABLED?, CUSTOM_WMI_QUERY_CHECKS?,
DO_NOT_OVERWRITE_OS?)>
<!ELEMENT FIM_CONTROLS_ENABLED (#PCDATA)>
<!ELEMENT CUSTOM_WMI_QUERY_CHECKS (#PCDATA)>
<!ELEMENT DO_NOT_OVERWRITE_OS (#PCDATA)>
...
<!ELEMENT IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK (#PCDATA)>
<!ELEMENT NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY (#PCDATA)>
```

Disable overriding OS value in subsequent scans

API affected	/api/2.0/fo/subscription/option_profile/
New or Updated API	Updated
DTD or XSD changes	Yes

You can configure an option profile to not override the OS detected by a previous scan. This is especially useful if you're running a light or custom scan and you don't want to overwrite the OS detected by a Full scan.

You need to set this option in the Qualys Cloud Platform UI. This option is available at the bottom of the Scan tab in the Option Profile wizard.



Once you set the option in the UI, the Option Profile - Export API will return the value as part of the XML response.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl"
-X GET "action=export&option_profile_id=130475"
"https://qualysapi.qualys.com/api/2.0/fo/subscription/option_profile/"
```

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>130475</ID>
     <GROUP_NAME><![CDATA[QRDI_Complete_QRDI_Disabled]]></GROUP_NAME>
     <GROUP_TYPE>user</GROUP_TYPE>
     <USER_ID><![CDATA[james smith]></USER_ID>
     <UNIT_ID>0</UNIT_ID>
     <SUBSCRIPTION_ID>45</SUBSCRIPTION_ID>
     <IS_DEFAULT>0</IS_DEFAULT>
     <IS_GLOBAL>1</IS_GLOBAL>
     <IS_OFFLINE_SYNCABLE>0</IS_OFFLINE_SYNCABLE>
     <UPDATE_DATE>2017-10-03T09:13:18Z</UPDATE_DATE>
   </BASIC_INFO>
   <SCAN>
    ...
   <DO_NOT_OVERWRITE_OS>1</DO_NOT_OVERWRITE_OS>
   </SCAN>
 </OPTION_PROFILE>
</OPTION_PROFILES>
```

To toggle the **Do not overwrite OS** option via API, you can import the option profile XML by including the following element info. Set the value 1 to enable the option.

```
<DO_NOT_OVERWRITE_OS>1</DO_NOT_OVERWRITE_OS>
```

DTD update:

The Option Profile Info DTD (option_profile_info.dtd) was updated to include the new element (in bold).

```
<!ELEMENT OPTION_PROFILES (OPTION_PROFILE)*>
<!ELEMENT OPTION_PROFILE (BASIC_INFO, SCAN, MAP?, ADDITIONAL)>
<!ELEMENT CONTROL_TYPES (FIM_CONTROLS_ENABLED?, CUSTOM_WMI_QUERY_CHECKS?,
DO_NOT_OVERWRITE_OS?)>
<!ELEMENT FIM_CONTROLS_ENABLED (#PCDATA)>
<!ELEMENT CUSTOM_WMI_QUERY_CHECKS (#PCDATA)>
<!ELEMENT DO_NOT_OVERWRITE_OS (#PCDATA)>

<!ELEMENT MAP (BASIC_INFO_GATHERING_ON, TCP_PORTS?, UDP_PORTS?,
MAP_OPTIONS?, MAP_PERFORMANCE?, MAP_AUTHENTICATION?)>

<!ELEMENT BASIC_INFO_GATHERING_ON (#PCDATA)>
```

Excluded Hosts List API - New tag filters

API affected	api/2.0/fo/asset/excluded_ip/
New or Updated API	Updated
DTD or XSD changes	No

We now added new filters to Excluded Hosts API (...api/2.0/fo/asset/excluded_ip/) for you to list excluded hosts that user has access to.

We have also enhanced the behavior of the network ID parameter. Let us consider different user scenarios:

User	Networks with access	Is network_id mandatory?	What does output include?
User 1	Global Default Network, Network 1, Network 2	No	Excluded host list from all the networks the user has access to.
User 2	Global Default Network	No	Excluded host list for global default network.
User 3	Network 1	Yes	Excluded host list for Network 1.
User 4	Network 1, Network 2, Network 3	Yes	Excluded host list for network that is listed in the request. Multiple entries are comma separated (for example, Network+1,Network+2,Network+3).

You could now user various filters that we support.

Parameter	Description
action=list	(Required) A flag used to make an excluded hosts list request.
Asset Groups	
ag_ids={value}	(Optional) Show excluded hosts belonging to asset groups with certain IDs. One or more asset group IDs and/or ranges may be specified. Multiple entries are comma separated. A range is specified with a dash (for example, 386941-386945). Valid asset group IDs are required.
	These parameters are mutually exclusive and cannot be specified together: ag_ids and ag_titles.

Parameter	Description
ag_titles={value}	(Optional) Show excluded hosts belonging to asset groups with certain strings in the asset group title. One or more asset group titles may be specified. Multiple entries are comma separated (for example, My+First+Asset+Group,Another+Asset+Group).
	These parameters are mutually exclusive and cannot be specified together: ag_ids and ag_titles.
Asset Tags	
use_tags={0|1}	(Optional) Specify 0 (the default) if you want to select hosts based on IP addresses/ranges and/or asset groups. Specify 1 if you want to select hosts based on asset tags.
tag_include_selector={any|all}	(Optional when use_tags=1) Specify "any" (the default) to include excluded hosts that match at least one of the selected tags. Specify "all" to include excluded hosts that match all of the selected tags.
tag_exclude_selector={any|all}	(Optional when use_tags=1) Specify "any" (the default) to ignore excluded hosts that match at least one of the selected tags. Specify "all" to ignore excluded hosts that match all of the selected tags.
tag_set_by = {id|name}	(Optional when use_tags=1) Specify "id" (the default) to select a tag set by providing tag IDs. Specify "name" to select a tag set by providing tag names.
tag_set_include={value}	(Optional when use_tags=1) Specify a tag set to include. Excluded hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
tag_set_exclude={value}	(Optional when use_tags=1) Specify a tag set to exclude. Excluded hosts that match these tags will be ignored. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.

Examples

Example 1: Using multiple tags to filter excluded host list

API request:

```
curl -u "USER:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d "content-
type: application/x-www-form-urlencoded"
"https://qualysapi.qualys.com/action=list&use_tags=1&tag_set_include=WIND
OWS_machine,UBUNTU_machine&tag_include_selector=all&tag_set_exclude=MAC_m
achine&tag_set_by=name"
```

XML output:

```
<!DOCTYPE IP_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/ip_list_output
.dtd">
<IP_LIST_OUTPUT>
   <REQUEST>
       <DATETIME>2017-10-03T10:02:20Z</DATETIME>
       <USER_LOGIN>user_patrick</USER_LOGIN>
<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/</RES
OURCE>
       <PARAM_LIST>
           <PARAM>
               <KEY>echo_request</KEY>
               <VALUE>1</VALUE>
           </PARAM>
           <PARAM>
               <KEY>use_tags</KEY>
               <VALUE>1</VALUE>
           </PARAM>
           <PARAM>
               <KEY>tag_include_selector</KEY>
               <VALUE>all</VALUE>
           </PARAM>
           <PARAM>
               <KEY>tag_set_include</KEY>
               <VALUE>WINDOWS_machine,UBUNTU_machine</VALUE>
           </PARAM>
           <PARAM>
               <KEY>tag_set_exclude</KEY>
               <VALUE>MAC_machine</VALUE>
           </PARAM>
           <PARAM>
               <KEY>tag_set_by</KEY>
               <VALUE>name</VALUE>
           </PARAM>
           <PARAM>
               <KEY>action</KEY>
               <VALUE>list</VALUE>
           </PARAM>
       </PARAM_LIST>
   </REQUEST>
   <RESPONSE>
       <DATETIME>2017-10-03T10:02:21Z</DATETIME>
       <IP_SET>
           <IP>10.10.36.63</IP>
       </IP_SET>
   </RESPONSE>
</IP_LIST_OUTPUT>
```

VM - Get additional information for detection type INFO

API affected	/api/2.0/fo/asset/host/vm/detection/
New or Updated API	Updated
DTD or XSD changes	No

The Host List Detection (.../api/2.0/fo/asset/host/vm/detection/) API now provides following additional information for the detection type "Info".

- severity level
- date and time when first detected
- date and time when last detected
- number of times detected

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample"
-d "action=list&ids=133024&show_igs=1"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_VM_DETECTION_OUTPUT SYSTEM
"http://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/host_list
_vm_detection_output.dtd">
<HOST_LIST_VM_DETECTION_OUTPUT>
    <RESPONSE>
        <DATETIME>2017-09-14T05:50:02Z</DATETIME>
        <HOST_LIST>
            <HOST>
                <ID>133024</ID>
                <IP>10.10.10.4</IP>
                <TRACKING_METHOD>IP</TRACKING_METHOD>
                <NETWORK_ID>0</NETWORK_ID>
                <OS>
                    <![CDATA[AIX 5.3]]>
                </OS>
                <DNS>
                    <![CDATA[10-10-10-4.bogus.tld]]>
                </DNS>
                <LAST_SCAN_DATETIME>2017-05-
                 26T09:32:20Z</LAST_SCAN_DATETIME>
                <LAST_VM_SCANNED_DATE>2017-05-
                 26T09:40:25Z</LAST_VM_SCANNED_DATE>
                <LAST_VM_SCANNED_DURATION>105</LAST_VM_SCANNED_DURATION>
```

```
                <LAST_VM_AUTH_SCANNED_DATE>2017-05-
                 26T09:40:25Z</LAST_VM_AUTH_SCANNED_DATE>
<LAST_VM_AUTH_SCANNED_DURATION>105</LAST_VM_AUTH_SCANNED_DURATION>
                <DETECTION_LIST>
                    <DETECTION>
                        <QID>6</QID>
                        <TYPE>Info</TYPE>
                        <SEVERITY>1</SEVERITY>
                        <RESULTS>
                            <![CDATA[IP addressHost name
                             10.10.10.410-10-10-4.bogus.tld]]>
                        </RESULTS>
                        <FIRST_FOUND_DATETIME>2017-03-
                         01T07:30:31Z</FIRST_FOUND_DATETIME>
                        <LAST_FOUND_DATETIME>2017-05-
                         26T09:40:25Z</LAST_FOUND_DATETIME>
                        <TIMES_FOUND>8</TIMES_FOUND>
                        <IS_DISABLED>0</IS_DISABLED>
                    </DETECTION>
                    <DETECTION>
                        <QID>9</QID>
                        <TYPE>Info</TYPE>
                        <SEVERITY>2</SEVERITY>
                        <PORT>111</PORT>
                        <PROTOCOL>tcp</PROTOCOL>
                        <RESULTS>
                            <![CDATA[programversionprotocolportname
                             1000004udp111rpcbind
                             1000831tcp32770ttdbserverd
                             1000682udp32772cmsd
                             1000211udp32774nlockmgr
                             2000012udp51385PyramidSys5]]>
                        </RESULTS>
                        <FIRST_FOUND_DATETIME>2017-03-
                         01T07:30:31Z</FIRST_FOUND_DATETIME>
                        <LAST_FOUND_DATETIME>2017-05-
                         26T09:40:25Z</LAST_FOUND_DATETIME>
                        <TIMES_FOUND>8</TIMES_FOUND>
                        <IS_DISABLED>0</IS_DISABLED>
                    </DETECTION>
                </DETECTION_LIST>
            </HOST>
        </HOST_LIST>
    </RESPONSE>
</HOST_LIST_VM_DETECTION_OUTPUT>
```

DTD:

No DTD changes

VM - Show QG Host ID for assets scanned with Agentless Tracking

API affected	/api/2.0/fo/asset/host/vm/detection/
New or Updated API	Updated
DTD or XSD changes	No
API affected	/api/2.0/fo/asset/host/
New or Updated API	Updated
DTD or XSD changes	No
API affected	N/A (affects Asset Data Report)
New or Updated API	No
DTD or XSD changes	No

You'll now see the QG Host ID (Qualys Host ID) for assets scanned with Agentless Tracking enabled (an option that allows you to track hosts by host ID). Previously the QG Host ID only appeared for assets with cloud agents installed.

Host List Detection API

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
"action=list&ips=10.0.203.170"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_VM_DETECTION_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/host_lis
t_vm_detection_output.dtd">
<HOST_LIST_VM_DETECTION_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-10-04T18:33:42Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>361030</ID>
        <IP>10.20.32.239</IP>
        <TRACKING_METHOD>IP</TRACKING_METHOD>
        <NETWORK_ID>0</NETWORK_ID>
        <OS><![CDATA[CentOS Linux 7.3.1611]]></OS>
```

```
<OS_CPE><![CDATA[cpe:/o:centos:centos_linux:7.3.1611:::]]></OS_CPE>
        <QG_HOSTID><![CDATA[58ca188c-01fc-0002-ad25-
a0423f216462]]></QG_HOSTID>
        <LAST_SCAN_DATETIME>2017-10-03T20:15:19Z</LAST_SCAN_DATETIME>
        <LAST_VM_SCANNED_DATE>2017-10-03T20:12:38Z</LAST_VM_SCANNED_DATE>
        <LAST_VM_SCANNED_DURATION>411</LAST_VM_SCANNED_DURATION>
        <LAST_VM_AUTH_SCANNED_DATE>2017-10-
03T20:12:38Z</LAST_VM_AUTH_SCANNED_DATE>
       <LAST_VM_AUTH_SCANNED_DURATION>411</LAST_VM_AUTH_SCANNED_DURATION>
        <LAST_PC_SCANNED_DATE>2017-09-18T19:05:13Z</LAST_PC_SCANNED_DATE>
        <DETECTION_LIST>
          <DETECTION>
            <QID>11</QID>
            <TYPE>Confirmed</TYPE>
            <SEVERITY>2</SEVERITY>
            <SSL>0</SSL>
           <RESULTS><![CDATA[Name     Program Version Protocol       Port
portmap/rpcbind 100000  2-4     tcp     111
portmap/rpcbind 100000  2-4     udp     905
portmap/rpcbind 100000  2-4     udp     111]]></RESULTS>
            <STATUS>Active</STATUS>
            <FIRST_FOUND_DATETIME>2017-08-
29T22:58:29Z</FIRST_FOUND_DATETIME>
            <LAST_FOUND_DATETIME>2017-10-
03T20:12:38Z</LAST_FOUND_DATETIME>
            <TIMES_FOUND>35</TIMES_FOUND>
            <LAST_TEST_DATETIME>2017-10-03T20:12:38Z</LAST_TEST_DATETIME>
            <LAST_UPDATE_DATETIME>2017-10-
03T20:15:19Z</LAST_UPDATE_DATETIME>
            <IS_IGNORED>0</IS_IGNORED>
            <IS_DISABLED>0</IS_DISABLED>
            <LAST_PROCESSED_DATETIME>2017-10-
03T20:15:19Z</LAST_PROCESSED_DATETIME>
          </DETECTION>
```

Host List API

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X -d
"action=list&details=All&ips=10.20.32.239"
"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/host_list_output.dtd"
```

```
>
<HOST_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-10-04T18:38:04Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>361030</ID>
        <IP>10.20.32.239</IP>
        <TRACKING_METHOD>IP</TRACKING_METHOD>
        <NETWORK_ID>0</NETWORK_ID>
        <OS><![CDATA[CentOS Linux 7.3.1611]]></OS>
        <QG_HOSTID><![CDATA[58ca188c-01fc-0002-ad25-
a0423f216462]]></QG_HOSTID>
        <LAST_VULN_SCAN_DATETIME>2017-10-
03T20:12:38Z</LAST_VULN_SCAN_DATETIME>
        <LAST_VM_SCANNED_DATE>2017-10-03T20:12:38Z</LAST_VM_SCANNED_DATE>
        <LAST_VM_SCANNED_DURATION>411</LAST_VM_SCANNED_DURATION>
        <LAST_VM_AUTH_SCANNED_DATE>2017-10-
03T20:12:38Z</LAST_VM_AUTH_SCANNED_DATE>
       <LAST_VM_AUTH_SCANNED_DURATION>411</LAST_VM_AUTH_SCANNED_DURATION>
        <LAST_COMPLIANCE_SCAN_DATETIME>2017-09-
18T19:05:13Z</LAST_COMPLIANCE_SCAN_DATETIME>
      </HOST>
    </HOST_LIST>
  </RESPONSE>
</HOST_LIST_OUTPUT>
```

Asset Data Report Update

You'll also see the QG Host ID when you fetch/download saved reports (in any format).

Example: Download report in CSV format

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X -d
"action=fetch&id=254606"
"https://qualysapi.qualys.com/api/2.0/fo/report/">rep1.csv
```

CSV output:

```
"qg_hostid report - CSV","10/05/2017 at 13:07:16 (GMT-0700)"
"qualys","123 first st",,"Redwood City","California","United States of
America","95131"
"user one","seenu_un","Unit Manager"

"Asset Groups","IPs","Active Hosts","Hosts Matching Filters","Trend
Analysis","Date Range","Network","Asset Tags"
"NONE","10.20.32.239","1","1","Latest vulnerability data","12/31/1998 -
```

```
10/05/2017","All","NONE"

"Total Vulnerabilities","Avg Security Risk","Business Risk"
"203","3.6","27/100"

"IP","Network","Total Vulnerabilities","Security Risk"
"10.20.32.239","Global Default Network","203","3.6"


"IP","Network","DNS","NetBIOS","QG Host ID","IP Interfaces","Tracking
Method","OS","IP Status","QID","Title","Vuln
Status","Type","Severity","Port","Protocol","FQDN","SSL","First
Detected","Last Detected","Times Detected","Date Last Fixed","First
Reopened","Last Reopened","Times Reopened","CVE ID","Vendor
Reference","Bugtraq
ID","Threat","Impact","Solution","Exploitability","Associated
Malware","Results","PCI Vuln","Ticket State","Instance","OS
CPE","Category"
"10.20.32.239","Global Default Network",,,"58ca188c-01fc-0002-ad25-
a0423f216462",,"IP","CentOS Linux 7.3.1611","host scanned, found
vuln","256248","CentOS Security Update for java-1.8.0-openjdk (CESA-
2017:1789)","Active","Vuln","5",,,,,"09/05/2017 15:59:31","10/05/2017
11:18:50","25",,,,,"CVE-2017-10053, CVE-2017-10067, CVE-2017-10074, CVE-
2017-10078, CVE-2017-10081, CVE-2017-10087, CVE-2017-10089, CVE-2017-
10090, CVE-2017-10096, CVE-2017-10101, CVE-2017-10102, CVE-2017-10107,
CVE-2017-10108, CVE-2017-10109, CVE-2017-10110, CVE-2017-10111, CVE-2017-
10115, CVE-2017-10116, CVE-2017-10135, CVE-2017-10193, CVE-2017-
10198","CESA-2017:1789 centos 6, CESA-2017:1789 centos 7","99842, 99756,
99731, 99752, 99853, 99703, 99659, 99706, 99670, 99674, 99712, 99719,
99846, 99847, 99643, 99707, 99774, 99734, 99839, 99854, 99818","CentOS has
released security update for java-1.8.0-openjdk to fix the
vulnerabilities. Affected Products:  centos 6 centos 7","Successful
exploitation allows attackers to compromise the system.","To resolve this
issue, upgrade to the latest packages which contain a patch. Refer to
CentOS advisory centos 6 (https://lists.centos.org/pipermail/centos-
announce/2017-July/022508.html) centos 7
(https://lists.centos.org/pipermail/centos-announce/2017-
July/022509.html) for updates and patch information.
 Patch:
Following are links for downloading patches to fix the vulnerabilities:
  CESA-2017:1789: centos 6 (https://lists.centos.org/pipermail/centos-
announce/2017-July/022508.html)  CESA-2017:1789: centos 7
(https://lists.centos.org/pipermail/centos-announce/2017-
July/022509.html)",,,"Package  Installed Version       Required Version
java-1.8.0-openjdk      1.8.0.102-4.b14.el7.x86_64      1.8.0.141-
1.b16.el7_3
java-1.8.0-openjdk-headless     1.8.0.102-4.b14.el7.x86_64      1.8.0.141-
1.b16.el7_3#","yes","Open",,"cpe:/o:centos:centos_linux:7.3.1611:::","Cen
tOS"
```

Example: Download report in XML format

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl" -X -d
"action=fetch&id=254606"
"https://qualysapi.qualys.com/api/2.0/fo/report/">rep1.xml
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE ASSET_DATA_REPORT SYSTEM
"https://qualysguard.qualys.com/asset_data_report.dtd">
<ASSET_DATA_REPORT>
  <HEADER>
    <COMPANY><![CDATA[qualys]]></COMPANY>
    <USERNAME>seenu_un</USERNAME>
    <GENERATION_DATETIME>2017-10-05T20:12:25Z</GENERATION_DATETIME>
    <TEMPLATE><![CDATA[qg_hostid template]]></TEMPLATE>
    <TARGET>
      <USER_IP_LIST>
        <RANGE network_id="-100">
          <START>10.20.32.239</START>
          <END>10.20.32.239</END>
        </RANGE>
      </USER_IP_LIST>
      <COMBINED_IP_LIST>
        <RANGE network_id="-100">
          <START>10.20.32.239</START>
          <END>10.20.32.239</END>
        </RANGE>
      </COMBINED_IP_LIST>
    </TARGET>
    <RISK_SCORE_SUMMARY>
      <TOTAL_VULNERABILITIES>203</TOTAL_VULNERABILITIES>
      <AVG_SECURITY_RISK>3.6</AVG_SECURITY_RISK>
      <BUSINESS_RISK>27/100</BUSINESS_RISK>
    </RISK_SCORE_SUMMARY>
  </HEADER>
  <RISK_SCORE_PER_HOST>
    <HOSTS>
      <IP_ADDRESS network_id="0">10.20.32.239</IP_ADDRESS>
      <TOTAL_VULNERABILITIES>203</TOTAL_VULNERABILITIES>
      <SECURITY_RISK>3.6</SECURITY_RISK>
    </HOSTS>
  </RISK_SCORE_PER_HOST>
  <HOST_LIST>
```

```
    <HOST>
      <IP network_id="0">10.20.32.239</IP>
      <TRACKING_METHOD>IP</TRACKING_METHOD>
      <QG_HOSTID><![CDATA[58ca188c-01fc-0002-ad25-
a0423f216462]]></QG_HOSTID>
      <OPERATING_SYSTEM><![CDATA[CentOS Linux
7.3.1611]]></OPERATING_SYSTEM>
      <OS_CPE><![CDATA[cpe:/o:centos:centos_linux:7.3.1611:::]]></OS_CPE>
      <ASSET_GROUPS>
        <ASSET_GROUP_TITLE><![CDATA[Mongo DB - AG]]></ASSET_GROUP_TITLE>
<ASSET_GROUPS>
        <ASSET_GROUP_TITLE><![CDATA[Mongo DB - AG]]></ASSET_GROUP_TITLE>
      </ASSET_GROUPS>
      <VULN_INFO_LIST>
        <VULN_INFO>
          <QID id="qid_42017">42017</QID>
          <TYPE>Ig</TYPE>
          <SSL>false</SSL>
          <RESULT><![CDATA[Service name: SSH on TCP port 22.]]></RESULT>
          <FIRST_FOUND>2017-08-29T22:58:29Z</FIRST_FOUND>
          <LAST_FOUND>2017-10-05T18:18:50Z</LAST_FOUND>
          <TIMES_FOUND>36</TIMES_FOUND>
        </VULN_INFO>
        <VULN_INFO>
          <QID id="qid_82046">82046</QID>
          <TYPE>Ig</TYPE>
          <SSL>false</SSL>
          <RESULT><![CDATA[IP ID changes observed (network order) for port
111: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Duration: 7 milli seconds]]></RESULT>
          <FIRST_FOUND>2017-08-29T22:58:29Z</FIRST_FOUND>
          <LAST_FOUND>2017-10-05T18:18:50Z</LAST_FOUND>
          <TIMES_FOUND>36</TIMES_FOUND>
        </VULN_INFO>
        <VULN_INFO>
          <QID id="qid_82063">82063</QID>
          <TYPE>Ig</TYPE>
          <SSL>false</SSL>
          <RESULT><![CDATA[Based on TCP timestamps obtained via port 111,
the host&apos;s uptime is 12 days, 20 hours, and 31 minutes.
The TCP timestamps from the host are in units of 1
milliseconds.]]></RESULT>
          <FIRST_FOUND>2017-08-29T22:58:29Z</FIRST_FOUND>
          <LAST_FOUND>2017-10-05T18:18:50Z</LAST_FOUND>
          <TIMES_FOUND>36</TIMES_FOUND>
        </VULN_INFO>
```

VM - Show QID Changes in KnowledgeBase API

API affected	/api/2.0/fo/knowledge_base/vuln/
New or Updated API	Updated
DTD or XSD changes	Yes

You'll now be able to view a list of changes made by Qualys to any QID in the Vulnerability KnowledgeBase including changes to detection logic, severity level and vulnerability type (confirmed, potential, information gathered). For each change you'll see the date of the change and comments provided by the Qualys Vulnerability Signatures team.

Good to Know - We will not display changes to QIDs made prior to this release. Only new changes will be recorded.

Input Parameters

Use the new parameter "show_qid_change_log" when requesting a list of vulnerabilities from the KnowledgeBase.

Parameter	Description
show_qid_change_log={**0**|1}	(Optional) Specify show_qid_change_log=1 to show QID changes in the XML output. When not specified, QID changes are not included in the output.

Examples

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=list&show_qid_change_log=1&ids=87290"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/"
```

XML output:

```
...
<CHANGE_LOG_LIST>
    <CHANGE_LOG_INFO>
        <CHANGE_DATE>
            <![CDATA[2017-09-16T12:21:04Z]]>
        </CHANGE_DATE>
        <COMMENTS>
            <![CDATA[Changed the severity from 4 to 5.]]>
        </COMMENTS>
    </CHANGE_LOG_INFO>
```

```
    <CHANGE_LOG_INFO>
        <CHANGE_DATE>
            <![CDATA[2017-09-12T04:15:05Z]]>
        </CHANGE_DATE>
        <COMMENTS>
            <![CDATA[Updated detection logic.]]>
        </COMMENTS>
    </CHANGE_LOG_INFO>
    <CHANGE_LOG_INFO>
        <CHANGE_DATE>
            <![CDATA[2017-05-16T10:11:09Z]]>
        </CHANGE_DATE>
        <COMMENTS>
            <![CDATA[Changed from Potential to Confirmed.]]>
        </COMMENTS>
    </CHANGE_LOG_INFO>
</CHANGE_LOG_LIST>
...
```

DTD update:

We added new elements (in bold) to the KnowledgeBase Output DTD (knowledge_base_vuln_list_output.dtd).

```
<!-- QUALYS KNOWLEDGE_BASE_VULN_LIST_OUTPUT DTD -->
...
    <!ELEMENT VULN_LIST (VULN*)>
      <!ELEMENT VULN (QID, VULN_TYPE, SEVERITY_LEVEL, TITLE, CATEGORY?,
DETECTION_INFO?, LAST_CUSTOMIZATION?,
LAST_SERVICE_MODIFICATION_DATETIME?, PUBLISHED_DATETIME,
BUGTRAQ_LIST?, PATCHABLE, SOFTWARE_LIST?, VENDOR_REFERENCE_LIST?,
CVE_LIST?, DIAGNOSIS?, DIAGNOSIS_COMMENT?, CONSEQUENCE?,
CONSEQUENCE_COMMENT?, SOLUTION?, SOLUTION_COMMENT?, COMPLIANCE_LIST?,
CORRELATION?, CVSS?, CVSS_V3?, PCI_FLAG?, AUTOMATIC_PCI_FAIL?,
PCI_REASONS?, THREAT_INTELLIGENCE?, SUPPORTED_MODULES?, DISCOVERY,
IS_DISABLED?, CHANGE_LOG_LIST? )>

...

        <!ELEMENT CHANGE_LOG_LIST (CHANGE_LOG_INFO+)>
          <!ELEMENT CHANGE_LOG_INFO (CHANGE_DATE, COMMENTS)>
            <!ELEMENT CHANGE_DATE (#PCDATA)>
            <!ELEMENT COMMENTS (#PCDATA)>
...
```

PC - View Asset Groups and Tag Information in XML Report

API affected	N/A (affects end report)
New or Updated API	No
DTD or XSD changes	Yes

The Compliance Policy Report DTD is now updated so that the policy report (xml) provides information about Asset Groups, IPs, Host Instances and Tags.

Updated Compliance Policy Report DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS COMPLIANCE POLICY REPORT DTD -->
<!-- $Revision$ -->

<!ELEMENT COMPLIANCE_POLICY_REPORT (ERROR | (HEADER, (SUMMARY),
(RESULTS)))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO,
FILTERS)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>

<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>

<!ELEMENT FILTERS (POLICY, POLICY_LOCKING?, ASSET_GROUPS?, IPS?,
HOST_INSTANCE?, ASSET_TAGS?, PC_AGENT_IPS?, POLICY_LAST_EVALUATED)>
<!ELEMENT POLICY (#PCDATA)>
<!ELEMENT POLICY_LOCKING (#PCDATA)>

<!ELEMENT ASSET_GROUPS (ASSET_GROUP?)>
<!ELEMENT ASSET_GROUP (ID, NAME)>
```

```
<!ELEMENT IPS (IP_LIST?, NEWWORK?)>
<!ELEMENT IP_LIST (IP)>
<!ELEMENT NEWWORK (#PCDATA)>

<!ELEMENT INCLUDED_TAGS (SCOPE, TAGS)>
<!ELEMENT EXCLUDED_TAGS (SCOPE, TAGS)>
<!ELEMENT TAGS (NAME*)>
<!ELEMENT SCOPE (#PCDATA)>

<!ELEMENT HOST_INSTANCE (IP?, INSTANCE?)>

<!ELEMENT PC_AGENT_IPS (#PCDATA)>

<!ELEMENT POLICY_LAST_EVALUATED (#PCDATA)>
<!ELEMENT SUMMARY (TOTAL_ASSETS, TOTAL_CONTROLS, CONTROL_INSTANCES,
CONTROLS_SUMMARY?, HOST_STATISTICS?)>
...
<!ELEMENT STATS (#PCDATA)>
<!ELEMENT SEARCH_DURATION (#PCDATA)>
<!ELEMENT ERRORS (#PCDATA)>
...

<!-- EOF -->
```

Sample XML output 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE COMPLIANCE_POLICY_REPORT SYSTEM
"https://qualysapi.qualys.com/compliance_policy_report.dtd">
<COMPLIANCE_POLICY_REPORT>
  <HEADER>
    <NAME><![CDATA[QWEB-14112]]></NAME>
    <GENERATION_DATETIME>2017-09-06T20:25:48Z</GENERATION_DATETIME>
    <COMPANY_INFO>
      <NAME><![CDATA[qualys]]></NAME>
      <ADDRESS><![CDATA[8,8]]></ADDRESS>
      <CITY><![CDATA[8]]></CITY>
      <STATE><![CDATA[Arkansas]]></STATE>
      <COUNTRY><![CDATA[United States of America]]></COUNTRY>
      <ZIP_CODE><![CDATA[88]]></ZIP_CODE>
    </COMPANY_INFO>
    <USER_INFO>
      <NAME><![CDATA[POC manager]]></NAME>
      <USERNAME>user_as</USERNAME>
      <ROLE>Manager</ROLE>
    </USER_INFO>
    <FILTERS>
      <POLICY><![CDATA[2 Controls ]]></POLICY>
      <POLICY_LOCKING><![CDATA[Unlocked]]></POLICY_LOCKING>
```

```
      <ASSET_GROUPS>
        <ASSET_GROUP>
          <ID><![CDATA[200425]]></ID>
          <NAME><![CDATA[Xp host with data]]></NAME>
        </ASSET_GROUP>
      </ASSET_GROUPS>
      <PC_AGENT_IPS><![CDATA[No]]></PC_AGENT_IPS>
      <POLICY_LAST_EVALUATED><![CDATA[08/23/2017 at 12:46:23 (GMT-
0700)]]></POLICY_LAST_EVALUATED>
    </FILTERS>
  </HEADER>
  ...
</COMPLIANCE_POLICY_REPORT>
```

Sample XML output 2:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE COMPLIANCE_POLICY_REPORT SYSTEM
"https://qualysapi.qualys.com/compliance_policy_report.dtd">
<COMPLIANCE_POLICY_REPORT>
  <HEADER>
    <NAME><![CDATA[QWEB-14112 tag]]></NAME>
    <GENERATION_DATETIME>2017-09-06T20:29:09Z</GENERATION_DATETIME>
    <COMPANY_INFO>
      <NAME><![CDATA[qualys]]></NAME>
      <ADDRESS><![CDATA[8,8]]></ADDRESS>
      <CITY><![CDATA[8]]></CITY>
      <STATE><![CDATA[Arkansas]]></STATE>
      <COUNTRY><![CDATA[United States of America]]></COUNTRY>
      <ZIP_CODE><![CDATA[88]]></ZIP_CODE>
    </COMPANY_INFO>
    <USER_INFO>
      <NAME><![CDATA[POC manager]]></NAME>
      <USERNAME>user</USERNAME>
      <ROLE>Manager</ROLE>
    </USER_INFO>
    <FILTERS>
      <POLICY><![CDATA[2 Controls ]]></POLICY>
      <POLICY_LOCKING><![CDATA[Unlocked]]></POLICY_LOCKING>
      <ASSET_TAGS>
        <INCLUDED_TAGS>
          <SCOPE><![CDATA[any]]></SCOPE>
          <TAGS>
            <NAME><![CDATA[Windows XP tag]]></NAME>
            <NAME><![CDATA[Windows 7]]></NAME>
          </TAGS>
        </INCLUDED_TAGS>
      </ASSET_TAGS>
      <PC_AGENT_IPS><![CDATA[No]]></PC_AGENT_IPS>
```

```
      <POLICY_LAST_EVALUATED><![CDATA[08/23/2017 at 12:46:23 (GMT-
0700)]]></POLICY_LAST_EVALUATED>
    </FILTERS>
  </HEADER>
  ...
 </COMPLIANCE_POLICY_REPORT>
```

PC - New UDC for Windows and Unix

Control List	
API affected	/api/2.0/fo/compliance/control/
New or Updated API	Updated
DTD or XSD changes	Yes
Compliance Policy Report	
API affected	N/A (affects end report)
New or Updated API	No
DTD or XSD changes	Yes

We have now updated Control API (.../api/2.0/fo/compliance/control) and Compliance Policy Report output to support integrity content check of Unix and Windows directory and files.

List Control API

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"https://qualysapi.qualys.com/api/2.0/fo/compliance/control/?action=list&
ids=100385"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/control/control_list_
output.dtd">
<CONTROL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2017-09-15T21:32:39Z</DATETIME>
    <CONTROL_LIST>
      <CONTROL>
        <ID>100385</ID>
        <UPDATE_DATE>2017-09-15T20:29:49Z</UPDATE_DATE>
        <CREATED_DATE>2017-09-15T20:28:36Z</CREATED_DATE>
        <CATEGORY>Anti-Virus/Malware</CATEGORY>
        <SUB_CATEGORY><![CDATA[Virus/Malware Prevention]]></SUB_CATEGORY>
        <STATEMENT><![CDATA[Unix Dir FIM UDC -String list]]></STATEMENT>
        <CRITICALITY>
          <LABEL><![CDATA[UNDEFINED]]></LABEL>
          <VALUE>0</VALUE>
        </CRITICALITY>
```

```
        <CHECK_TYPE><![CDATA[Unix Directory Integrity
Check]]></CHECK_TYPE>
        <COMMENT><![CDATA[]]></COMMENT>
        <IGNORE_ERROR>0</IGNORE_ERROR>
        <SCAN_PARAMETERS>
            <BASE_DIR><![CDATA[/usr]]></BASE_DIR>
            <SHOULD_DESCEND><![CDATA[true]]></SHOULD_DESCEND>

<INTEGRITY_CHECK_DEPTH_LIMIT><![CDATA[14]]></INTEGRITY_CHECK_DEPTH_LIMIT>
            <FOLLOW_SYMLINK><![CDATA[false]]></FOLLOW_SYMLINK>
            <FILE_NAME_MATCH><![CDATA[*conf]]></FILE_NAME_MATCH>
            <FILE_NAME_SKIP><![CDATA[]]></FILE_NAME_SKIP>
            <DIR_NAME_MATCH><![CDATA[/var]]></DIR_NAME_MATCH>
            <DIR_NAME_SKIP><![CDATA[]]></DIR_NAME_SKIP>
            <TYPE_MATCH><![CDATA[f,l,b,c]]></TYPE_MATCH>
            <USER_OWNER><![CDATA[Any User]]></USER_OWNER>
            <GROUP_OWNER><![CDATA[Any Group]]></GROUP_OWNER>

<INTEGRITY_CHECK_TIME_LIMIT><![CDATA[600]]></INTEGRITY_CHECK_TIME_LIMIT>

<INTEGRITY_CHECK_MATCH_LIMIT><![CDATA[512]]></INTEGRITY_CHECK_MATCH_LIMIT
>
            <DIGEST_HASH><![CDATA[SHA-1]]></DIGEST_HASH>
            <DATA_TYPE>String</DATA_TYPE>
            <DESCRIPTION><![CDATA[UDC testing]]></DESCRIPTION>
        </SCAN_PARAMETERS>
        <TECHNOLOGY_LIST>
          <TECHNOLOGY>
            <ID>52</ID>
            <NAME>AIX 7.x</NAME>
            <RATIONALE><![CDATA[test]]></RATIONALE>
            <DATAPOINT>
                <CARDINALITY>no cd</CARDINALITY>
                <OPERATOR>xeq</OPERATOR>
                <DEFAULT_VALUES total="1">

<DEFAULT_VALUE><![CDATA[USE_SCAN_VALUE]]></DEFAULT_VALUE>
                </DEFAULT_VALUES>
            </DATAPOINT>
            <USE_SCAN_VALUE>0</USE_SCAN_VALUE>
          </TECHNOLOGY>
        </TECHNOLOGY_LIST>
      </CONTROL>
    </CONTROL_LIST>
  </RESPONSE>
</CONTROL_LIST_OUTPUT>
```

DTD:

```
<!-- QUALYS CONTROL_LIST_OUTPUT DTD -->
<!ELEMENT CONTROL_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
POST_DATA?)>
...
<!ELEMENT COMMENT (#PCDATA)>
<!ELEMENT IGNORE_ERROR (#PCDATA)>
<!ELEMENT IGNORE_ITEM_NOT_FOUND (#PCDATA)>
<!ELEMENT SCAN_PARAMETERS (REG_HIVE?, REG_KEY?, REG_VALUE_NAME?,
FILE_PATH?, FILE_QUERY?, HASH_TYPE?, WMI_NS?, WMI_QUERY?, SHARE_USER?,
PATH_USER?, GROUP_NAME?, GROUP_NAME_LIMIT?,
BASE_DIR?, SHOULD_DESCEND?, DEPTH_LIMIT?, INTEGRITY_CHECK_DEPTH_LIMIT?,
FOLLOW_SYMLINK?, FILE_NAME_MATCH?, FILE_NAME_SKIP?, DIR_NAME_MATCH?,
DIR_NAME_SKIP?, WIN_FILE_SYS_OBJECT_TYPES?,
MATCH_WELL_KNOWN_USERS_FOR_ANY_DOMAIN?, WIN_PERMISSION_USERS?,
WIN_PERMISSION_MATCH?, WIN_PERMISSIONS?, PERMISSIONS?, PERM_COND?,
TYPE_MATCH?, USER_OWNER?, GROUP_OWNER?, TIME_LIMIT?, MATCH_LIMIT?,
INTEGRITY_CHECK_TIME_LIMIT?, INTEGRITY_CHECK_MATCH_LIMIT?, DIGEST_HASH?,
DATA_TYPE, DESCRIPTION)>
<!ELEMENT REG_HIVE (#PCDATA)>
<!ELEMENT REG_KEY (#PCDATA)>
...
<!ELEMENT BASE_DIR (#PCDATA)>
<!ELEMENT DEPTH_LIMIT (#PCDATA)>
<!ELEMENT INTEGRITY_CHECK_DEPTH_LIMIT (#PCDATA)>
<!ELEMENT FILE_NAME_MATCH (#PCDATA)>
<!ELEMENT FILE_NAME_SKIP (#PCDATA)>
...
<!ELEMENT READ (#PCDATA)>
<!ELEMENT WRITE (#PCDATA)>
<!ELEMENT EXECUTE (#PCDATA)>

<!ELEMENT INTEGRITY_CHECK_TIME_LIMIT (#PCDATA)>
<!ELEMENT INTEGRITY_CHECK_MATCH_LIMIT (#PCDATA)>
<!ELEMENT DIGEST_HASH (#PCDATA)>

<!ELEMENT DATA_TYPE (#PCDATA)>
...
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!-- EOF -->
```

Fetch Compliance Policy Report API

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample" -d
"action=fetch&ids=157126"
"https://qualysapi.qualys.com/api/2.0/fo/report/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE COMPLIANCE_POLICY_REPORT SYSTEM
"https://qualysapi.qualys.com/compliance_policy_report.dtd">
<COMPLIANCE_POLICY_REPORT>
  <HEADER>
    <NAME><![CDATA[Dir based FIM UDC XML]]></NAME>
    <GENERATION_DATETIME>2017-09-18T21:27:38Z</GENERATION_DATETIME>
    <COMPANY_INFO>
      <NAME><![CDATA[qualys]]></NAME>
      ...
      <CHECK>
        <NAME>CHECK2</NAME>
        <DP_NAME>custom.win_dir_integrity_check.1063032</DP_NAME
  ...
<ACTUAL lastUpdated="2017-09-18T18:54:20Z">

<V><![CDATA[c:\WINDOWS\dialer.exe|:|4609bb6e0ee01ed291a8f11e33495799|:|]
]></V>

<V><![CDATA[c:\WINDOWS\explorer.exe|:|ae7a08c05f72a9242734c03230a5cd7f|:
|]]></V>

<V><![CDATA[c:\WINDOWS\hh.exe|:|1bb94c8d6d32bb4782ce45d96ab7d79d|:|]]></
V>

<V><![CDATA[c:\WINDOWS\NOTEPAD.EXE|:|d7bfdb6d6acf2d7e0ecb25d2756d8fd6|:|
]]></V>

<V><![CDATA[c:\WINDOWS\regedit.exe|:|872a60b75ce6a09033fbe2461d44e696|:|
]]></V>

<V><![CDATA[c:\WINDOWS\splwow64.exe|:|3f3e904c7a57e3a14197192046851c87|:
|]]></V>

<V><![CDATA[c:\WINDOWS\twunk_16.exe|:|f36a271706edd23c94956afb56981184|:
|]]></V>

<V><![CDATA[c:\WINDOWS\twunk_32.exe|:|80ea8e830cad7c8fc2c288b5eb79dd11|:
|]]></V>
```

```
<V><![CDATA[c:\WINDOWS\winhlp32.exe|:|e5f713e5ea86e28274f8091465186545|:
|]]></V>
        </ACTUAL>
        <ADDED_DIRECTORIES />
        <REMOVED_DIRECTORIES>

<V><![CDATA[c:\WINDOWS\TASKMAN.EXE|:|f4dfd83153e8c9088ae2db704107060d|:|
]]></V>

<V><![CDATA[c:\WINDOWS\winhelp.exe|:|8e6f7d51a5cb299c25621c6c1ab57e84|:|
]]></V>
        </REMOVED_DIRECTORIES>
        <PERMISSON_CHANGED_DIRECTORIES>

<V><![CDATA[c:\WINDOWS\NOTEPAD.EXE|:|388b8fbc36a8558587afc90fb23a3b99|:|
]]></V>

<V><![CDATA[c:\WINDOWS\explorer.exe|:|a0732187050030ae399b241436565e64|:
|]]></V>

<V><![CDATA[c:\WINDOWS\hh.exe|:|de6fee4defbc2a7d54ac0227191f827e|:|]]></
V>

<V><![CDATA[c:\WINDOWS\regedit.exe|:|783afc80383c176b22dbf8333343992d|:|
]]></V>

<V><![CDATA[c:\WINDOWS\twunk_32.exe|:|a68224457dd43d18e40e02262d4a9398|:
|]]></V>

<V><![CDATA[c:\WINDOWS\winhlp32.exe|:|3371d02425bf6d8ca33de9c92f359519|:
|]]></V>
        </PERMISSON_CHANGED_DIRECTORIES>
        <CONTENT_CHANGED_DIRECTORIES>

<V><![CDATA[c:\WINDOWS\NOTEPAD.EXE|:|d7bfdb6d6acf2d7e0ecb25d2756d8fd6|:|
]]></V>

<V><![CDATA[c:\WINDOWS\dialer.exe|:|4609bb6e0ee01ed291a8f11e33495799|:|]
]></V>

<V><![CDATA[c:\WINDOWS\explorer.exe|:|ae7a08c05f72a9242734c03230a5cd7f|:
|]]></V>

<V><![CDATA[c:\WINDOWS\hh.exe|:|1bb94c8d6d32bb4782ce45d96ab7d79d|:|]]></
V>

<V><![CDATA[c:\WINDOWS\regedit.exe|:|872a60b75ce6a09033fbe2461d44e696|:|
]]></V>
```

```
<V><![CDATA[c:\WINDOWS\splwow64.exe|:|3f3e904c7a57e3a14197192046851c87|:
|]]></V>

<V><![CDATA[c:\WINDOWS\twunk_32.exe|:|80ea8e830cad7c8fc2c288b5eb79dd11|:
|]]></V>

<V><![CDATA[c:\WINDOWS\winhlp32.exe|:|e5f713e5ea86e28274f8091465186545|:
|]]></V>
        </CONTENT_CHANGED_DIRECTORIES>
        <EXTENDED_EVIDENCE><![CDATA[Row 1:Path,Size,Create time,Modify
time,Access time,Digest
...
</COMPLIANCE_POLICY_REPORT>
```

DTD:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS COMPLIANCE POLICY REPORT DTD -->
<!ELEMENT COMPLIANCE_POLICY_REPORT (ERROR | (HEADER, (SUMMARY),
(RESULTS)))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

...
<!ELEMENT RESULTS ( HOST_LIST, CHECKS?, DP_DESCRIPTIONS?) >
<!ELEMENT HOST_LIST (HOST*)>
<!ELEMENT HOST (TRACKING_METHOD, IP, DNS?, NETBIOS?, OPERATING_SYSTEM?,
OS_CPE?, LAST_SCAN_DATE?,TOTAL_PASSED, TOTAL_FAILED, TOTAL_ERROR,
TOTAL_EXCEPTIONS, ASSET_TAGS?, CONTROL_LIST, NETWORK?)>

<!ELEMENT CHECKS (CHECK*)>
<!ELEMENT CHECK (NAME, DP_NAME, EXPECTED, ACTUAL, ADDED_DIRECTORIES?,
REMOVED_DIRECTORIES?, PERMISSON_CHANGED_DIRECTORIES?,
CONTENT_CHANGED_DIRECTORIES?, PERMISSION_TRANSLATION?,
EXTENDED_EVIDENCE?, STATISTICS?)>
<!ELEMENT DP_NAME (#PCDATA)>
<!ELEMENT EXTENDED_EVIDENCE (#PCDATA)>
<!ELEMENT STATISTICS (STATS*, SEARCH_DURATION?, ERRORS?)>
<!ELEMENT EVALUATION (#PCDATA)>

<!ELEMENT EXPECTED (V*, CRITERIA?)>
<!ATTLIST EXPECTED logic CDATA #FIXED "OR">
<!ELEMENT CRITERIA (EVALUATION, V*)>
<!ELEMENT ACTUAL (V*)>
<!ELEMENT V (#PCDATA)>
<!ATTLIST ACTUAL lastUpdated CDATA #IMPLIED>

<!ELEMENT ADDED_DIRECTORIES (V*)>
<!ELEMENT REMOVED_DIRECTORIES (V*)>
```

```
<!ELEMENT PERMISSON_CHANGED_DIRECTORIES (V*)>
<!ELEMENT CONTENT_CHANGED_DIRECTORIES (V*)>

<!ELEMENT PERMISSION_TRANSLATION (PAIR+)>
<!ELEMENT PAIR (K, V)>
<!ELEMENT K (#PCDATA)>

...
<!ELEMENT STATS (#PCDATA)>
<!ELEMENT SEARCH_DURATION (#PCDATA)>
<!ELEMENT ERRORS (#PCDATA)>
```

New way to track API usage

You can now track API usage by a user without the need to provide user credentials such as the username and password.

API usage can be tracked using the X-Powered-By HTTP header which includes a unique ID generated for each subscription and a unique ID generated for each user. Once enabled, the X-Powered-By HTTP header is returned for each API request made by a user.

The information is returned in the following format:

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

Where,

POD_ID is the shared POD or a PCP. Shared POD is USPOD1, USPOD2, etc.

SUB_UUID is the unique ID generated for the subscription

USER_UUID is the unique ID generated for the user

For example,

```
X-Powered-By: Qualys:USPOD1:d9a7e94c-0a9d-c745-82e9-
980877cc5043:f178af1e-4049-7fce-81ca-75584feb8e93
```

You can use the USER_UUID to track API usage per user.

Contact Qualys Support to get the X-Powered-By HTTP header enabled.

Good to Know - This feature applies to Qualys APIs in Qualys API User Guide v1 and Qualys API User Guide v2 at this time. Other APIs documented in other API user guides will be updated in a future release.

Looking for our latest API user guides? Visit our Documentation page here: https://community.qualys.com/docs/DOC-4802

Sample output:

```
...
< HTTP/1.1 200 OK
< Date: Thu, 14 Sep 2017 09:11:21 GMT
< Server: Qualys
< X-XSS-Protection: 1
< X-Content-Type-Options: nosniff
< X-Frame-Options: SAMEORIGIN
< X-Powered-By: Qualys:USPOD1:d9a7e94c-0a9d-c745-82e9-
980877cc5043:f178af1e-4049-7fce-81ca-75584feb8e93
< X-RateLimit-Limit: 300
< X-RateLimit-Window-Sec: 3600
```

```
< X-Concurrency-Limit-Limit: 500
< X-Concurrency-Limit-Running: 0
< X-RateLimit-ToWait-Sec: 0
< X-RateLimit-Remaining: 298
< X-Qualys-Application-Version: QWEB-8.11.0.0-SNAPSHOT-
20170914072818#4205
< X-Server-Virtual-Host: qualysapi.p04.eng.qualys.com
< X-Server-Http-Host: qualysapi.p04.eng.qualys.com
< Transfer-Encoding: chunked
< Content-Type: text/xml;charset=UTF-8
...
```

The X-Powered-By HTTP header will be returned for both valid and invalid requests. However, it will not be returned when user authentication fails.