# Qualys API Release Notes

## Version 8.9

Qualys 8.9 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

**Tell me about the base URL** Our documentation and sample code use the API server URL for Qualys US Platform 1. Do you have another base URL? If yes please use it instead.

| Account Login | Base URL |
|---|---|
| Qualys US Platform 1 | https://qualysapi.qualys.com |
| Qualys US Platform 2 | https://qualysapi.qg2.apps.qualys.com |
| Qualys EU Platform | https://qualysapi.qualys.eu |
| Qualys Private Cloud Platform | https://qualysapi.<customer_base_url> |

# Unix Authentication Improvements

We're excited to tell you about the many enhancements we've made to Unix authentication in this release. All enhancements are available for the Unix Record using Qualys Cloud Platform UI and API. Now you can configure a single authentication record that supports better integration with third party vaults, and lets you define a variety of private keys and root delegation tools.

### What's New

Ability to get password for user login credentials from vault

Use multiple private-key certificates (RSA, DSA, ECDSA, ED25519)

Use multiple root delegation tools (Sudo, Pimsu, PowerBroker)

Private-key certificates NEW OPTIONS

- get private key from vault - CyberArk AIM vault only
- add vault user passphrase
- get passphrase from vault

Root delegation NEW OPTIONS

- get password from vault
- add vault user passphrase
- get passphrase from vault

### Your existing Unix records will be upgraded

We'll upgrade all of your Unix records to the new SSH2 authentication record and the upgraded records will function exactly as before and do not require any changes by you.

Good to Know - Cisco records and CheckPoint Firewall records will remain the same and will not be upgraded.

### What are the API changes?

To create, update, delete, list Unix records, use the same Unix Record endpoint (https://<qualysapi.qualys.com>/api/2.0/fo/auth/unix) to make requests. Define parameters for root delegations and private-key certificates in an XML file (unix_auth_params.dtd) and add to request parameters (i.e. create, update, delete).

Updated DTD for listing Unix records: auth_unix_list_output.dtd

New DTD for defining new Unix record parameters: unix_auth_params.dtd

## Unix Record Parameters

New Request Parameters:

| Parameter | Description |
|---|---|
| skip_password={**0**\|1} | (Optional) By default when only the required parameters are set (title, username, ips) the login account password is set to the empty password - the same behavior as in previous releases. Now you can set skip_password=1 if the login account does not have a password. When set it's not possible to set the empty password, another password using the "password" parameter, or password in a vault. |
| {XML File} | (Optional) XML file where you define private-key certificates and root delegations. |

These parameters are no longer supported to define an SSH2 authentication record: root_tool, enable_sudo, rsa_private_key, dsa_private_key. They are still supported for Cisco and CheckPoint Firewall records.

New XML File with parameters:

Starting with this release you'll configure private-key certifications and root delegations for Unix records in an XML file:
https://<platformURL>/api/2.0/fo/auth/unix/unix_auth_params.dtd

```
<!-- QUALYS UNIX_AUTH_PARAMS DTD -->
<!ELEMENT UNIX_AUTH_PARAMS (ROOT_TOOLS?, PRIVATE_KEY_CERTIFICATES?)>
<!ELEMENT ROOT_TOOLS (ROOT_TOOL)*>
<!ELEMENT ROOT_TOOL (ID?, (STANDARD_TYPE|CUSTOM_TYPE), PASSWORD_INFO)>
<!-- ID may not be specified for any applicable in edit mode-->
<!ELEMENT ID (#PCDATA)>
<!ELEMENT STANDARD_TYPE (#PCDATA)>
<!ATTLIST STANDARD_TYPE
        type (sudo|pimsu|powerbroker) #REQUIRED>
<!ELEMENT CUSTOM_TYPE (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT PASSWORD_INFO (DIGITAL_VAULT|PASSWORD)>
<!ATTLIST PASSWORD_INFO
        type (basic|vault) #REQUIRED>
<!ELEMENT DIGITAL_VAULT (VAULT_INFO_ID?, VAULT_USERNAME?, VAULT_TYPE?,
VAULT_ID?, FOLDER?, FILE?,SECRET_NAME?, SYSTEM_NAME?, END_POINT_NAME?,
END_POINT_TYPE?, END_POINT_CONTAINER?, AUTO_DISCOVER_SYSTEM_NAME?,
SYSTEM_NAME_SINGLE_HOST?, SYSTEM_TYPE?, CUSTOM_SYSTEM_TYPE?)>
<!-- VAULT_USERNAME may ONLY be used if used within PASSPHRASE_INFO or
PASSWORD_INFO -->
<!ELEMENT VAULT_INFO_ID (#PCDATA)>
<!ELEMENT VAULT_USERNAME (#PCDATA)>
<!ELEMENT VAULT_TYPE (#PCDATA)>
```

```
<!ELEMENT VAULT_ID (#PCDATA)>
<!-- Cyber-Ark PIM Suite/ Cyber-Ark AIM -->
<!ELEMENT FOLDER (#PCDATA)>
<!ELEMENT FILE (#PCDATA)>
<!-- -->
<!-- Thycotic Secret Server -->
<!ELEMENT SECRET_NAME (#PCDATA)>
<!-- -->
<!-- Quest Vault -->
<!ELEMENT SYSTEM_NAME (#PCDATA)>
<!-- -->
<!-- CA Access Control -->
<!ELEMENT END_POINT_NAME (#PCDATA)>
<!ELEMENT END_POINT_TYPE (#PCDATA)>
<!ELEMENT END_POINT_CONTAINER (#PCDATA)>
<!-- -->
<!-- Lieberman ERPM -->
<!ELEMENT AUTO_DISCOVER_SYSTEM_NAME (#PCDATA)>
<!ELEMENT SYSTEM_NAME_SINGLE_HOST (#PCDATA)>
<!ELEMENT SYSTEM_TYPE (#PCDATA)>
<!ELEMENT CUSTOM_SYSTEM_TYPE (#PCDATA)>
<!-- -->
<!ELEMENT PASSWORD (#PCDATA)>
<!ELEMENT PRIVATE_KEY_CERTIFICATES (PRIVATE_KEY_CERTIFICATE)*>
<!ELEMENT PRIVATE_KEY_CERTIFICATE (ID?, PRIVATE_KEY_INFO,
PASSPHRASE_INFO?, CERTIFICATE?)>
<!ELEMENT PRIVATE_KEY_INFO (DIGITAL_VAULT|PRIVATE_KEY)>
<!ATTLIST PRIVATE_KEY_INFO
        type (basic|vault) #REQUIRED>
<!ELEMENT PASSPHRASE_INFO (PASSPHRASE|DIGITAL_VAULT)>
<!ATTLIST PASSPHRASE_INFO
        type (basic|vault) #REQUIRED>
<!ELEMENT PASSPHRASE (#PCDATA)>
<!ELEMENT CERTIFICATE (#PCDATA)>
<!ATTLIST CERTIFICATE
        type (x.509|openssh) #REQUIRED>
<!ELEMENT PRIVATE_KEY (#PCDATA)>
<!ATTLIST PRIVATE_KEY
        type (rsa|dsa|ecdsa|ed25519) #REQUIRED>
<!-- EOF -->
```

### Create Unix Record

API request 1:

Create a Unix record and add the password for login, without adding any root delegation tools or private-key certificates.

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=create&title=U
nix&username=root&password=crazy8!&ips=10.10.36.63"
```

API request 2:

Create a Unix record without adding any root delegation tools or private-key certificates AND set skip_password=1 if the login account does not have a password. (If this account has the empty password, just enter the required parameters title, username and ips as in previous releases and the empty password will be used for login).

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=create&title=U
nix&username=root&skip_password=1&ips=10.10.36.63"
```

API request 3:

Create a Unix record and add multiple root delegation tools and private-key certificates AND use the Lieberman ERPM vault for login.

```
curl -H "X-Requested-With: curl" -H "Content-type:text/xml" -u
"USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/action=create&title=Un
ix&vault&username=Qualys&ips=10.113.195.152&port=5857&login_type=vault&va
ult_type=LiebermanERPM&vault_id=10873203&auto_discover_system_name=0&syst
em_name_single_host=a&custom_system_type=custom&system_type=custom"
--data-binary @add_params.xml
```

File add_params.xml contains multiple root delegation tools and private-key certificates:

```
<?xml version="1.0" encoding="UTF-8" ?>
<UNIX_AUTH_PARAMS>
    <ROOT_TOOLS>
        <ROOT_TOOL>
            <STANDARD_TYPE type="pimsu"/>
            <PASSWORD_INFO type="vault">
                <DIGITAL_VAULT>
                    <VAULT_USERNAME><![CDATA[root]]></VAULT_USERNAME>
                    <VAULT_TYPE>Thycotic Secret Server</VAULT_TYPE>
                    <VAULT_ID>25026922</VAULT_ID>
<SECRET_NAME><![CDATA[super_secret_name]]></SECRET_NAME>
```

```
                    </DIGITAL_VAULT>
                </PASSWORD_INFO>
            </ROOT_TOOL>
            <ROOT_TOOL>
                <CUSTOM_TYPE><![CDATA[test]]></CUSTOM_TYPE>
                <PASSWORD_INFO type="basic">
                    <PASSWORD><![CDATA[password]]></PASSWORD>
                </PASSWORD_INFO>
            </ROOT_TOOL>
        </ROOT_TOOLS>
        <PRIVATE_KEY_CERTIFICATES>
            <PRIVATE_KEY_CERTIFICATE>
                <PRIVATE_KEY_INFO type="vault">
                    <DIGITAL_VAULT>
                        <VAULT_TYPE>Cyber-Ark AIM</VAULT_TYPE>
                        <VAULT_ID>25026922</VAULT_ID>
                        <FOLDER><![CDATA[folder]]></FOLDER>
                        <FILE><![CDATA[file]]></FILE>
                    </DIGITAL_VAULT>
                </PRIVATE_KEY_INFO>
                <PASSPHRASE_INFO type="basic">
                    <PASSPHRASE><![CDATA[passphrase]]></PASSPHRASE>
                </PASSPHRASE_INFO>
            </PRIVATE_KEY_CERTIFICATE>
            <PRIVATE_KEY_CERTIFICATE>
                <PRIVATE_KEY_INFO type="basic">
                    <PRIVATE_KEY type="rsa">
<![CDATA[-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,F9A653E2D12E019357B349B6EEE068B1

FiLfGHOc0rREmC0cBPsiyqqaitPNYTGeqKRmSBwGNrAzNTAcsKslsoY/WkMDW6QD
dLZNiGB0CFag94zyoMyCjyrdpayACAOWfH5w8VixxHF16Vxx5b6foLBE40FOYAIP
sdm1HvCfSFaN2dPf1Unb0erwjigjJNwYIV78529elE+2+dZIemi90ibh0R35NB60
TLeS3UUVezp/O9ZPLf0pqPPHnWgfW4GXp/SUpwojES9fCQE+BW4MMWHWu8XKtytt
....
-----END RSA PRIVATE KEY-----]]></PRIVATE_KEY>
                </PRIVATE_KEY_INFO>
                <PASSPHRASE_INFO type="vault">
                    <DIGITAL_VAULT>
                        <VAULT_USERNAME><![CDATA[PASSPHRASE
USERNAME]]></VAULT_USERNAME>
                        <VAULT_TYPE>Quest Vault</VAULT_TYPE>
                        <VAULT_ID>35046922</VAULT_ID>
                      <SYSTEM_NAME><![CDATA[quest_system_name]]></SYSTEM_NAME>
                    </DIGITAL_VAULT>
                </PASSPHRASE_INFO>
                <CERTIFICATE type="openssh">
                    <![CDATA[ssh-rsa-cert-v01@openssh.com
```

```
AAAAHHNzaC1yc2EtY2VydC12MDFAb3BlbnNzaC5jb20AAAAgwR4bJSiBtJlOgCAQUF3yZ6Io2
WYfnBiOEsQ45RKbqLgAAAADAQABAAABAQC5sVLb7emh8/v2uHp6x1pN5R+MHQwz3A5M3GRKtu
uu1Njc/XYgqeWLMOJpbVtCVXwUcPgKt4Q0DmlGqc4uhZhzrdtpQGHrEivndNNLY9NQj7LozE7
x/sGiWdtmlucUh1teXMaBpM4aER9Y6uW5wv6ZylY7CAV9bcVz/ljlSypmjzkPjJ39AJq+QxZk
Iv+H4uh/T05LwHdilFrjWWwEoI8DV/DRIw3h8o4jhnj1QxBxyjad3efmFaejgRnY6cBW821gm
...
            </CERTIFICATE>
        </PRIVATE_KEY_CERTIFICATE>
        <PRIVATE_KEY_CERTIFICATE>
            <PRIVATE_KEY_INFO type="basic">
                <PRIVATE_KEY type="rsa">
<![CDATA[-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jYmMAAAAGYmNyeXB0AAAAGAAAABCPiEUH5L
3LZGInEw+h/m4+AAAAEAAAAAEAAAEXAAAAB3NzaC1yc2EAAAADAQABAAABAQCpuwFVTYVm
ske0bdFjSlYgsfvyCr7e5irIfoW7B8hNY0XJWyOEqZ5BzwPAEtzjua6m3vnqKPEQD1HyFd
Lse62JE7x0jDXLr9bZ64THFpogERC/gI2aorrLKLxdr0K7u5wQUTm1L0xO7Y0hE9Bbi8ok
++xTW+Ymf7LbVRLWVdN6kUBunIGow3W+tHIohPoUlw82QayZRa4iXpqpWVbh/9OMnb1raC
....
-----END OPENSSH PRIVATE KEY-----]]></PRIVATE_KEY>
            </PRIVATE_KEY_INFO>
        </PRIVATE_KEY_CERTIFICATE>
    </PRIVATE_KEY_CERTIFICATES>
</UNIX_AUTH_PARAMS>"
```

## Edit Root Delegations and Private Keys

Use a Unix record update request including XML binary data like this:

```
curl -H "X-Requested-With: curl" -H "Content-type:text/xml"
-u "USERNAME:PASSWORD” -X "POST"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/action=update&id=12345
67" --data-binary @edit_params.xml
```

Edit root tools:

Root tools in file binary_input_params.xml will be added. Any existing root tools will be deleted from the Unix record.

where edit_params.xml is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<UNIX_AUTH_PARAMS>
    <ROOT_TOOLS>
        <ROOT_TOOL>
            <ID>140016922</ID>
            <STANDARD_TYPE type="pimsu"/>
            <PASSWORD_INFO type="vault">
                <DIGITAL_VAULT>
                    <VAULT_USERNAME><![CDATA[root]]></VAULT_USERNAME>
```

```
                    <VAULT_TYPE>Thycotic Secret Server</VAULT_TYPE>
                    <VAULT_ID>25026922</VAULT_ID>

<SECRET_NAME><![CDATA[super_secret_name]]></SECRET_NAME>
                </DIGITAL_VAULT>
            </PASSWORD_INFO>
        </ROOT_TOOL>
        <!-- in add_root_tools.xml we had created two root-tools; here we
are specifying only one item to edit, so the other record will be deleted!
-->
    </ROOT_TOOLS>
</UNIX_AUTH_PARAMS>
```

Edit private-key certificates:

Private-key certificates in file binary_input_params.xml will be added. Any existing private-key certificates will be deleted from the Unix record.

where edit_params.xml is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<UNIX_AUTH_PARAMS>
    <PRIVATE_KEY_CERTIFICATES>
        <PRIVATE_KEY_CERTIFICATE>
            <ID>110066922</ID>
            <PRIVATE_KEY_INFO type="vault">
                <DIGITAL_VAULT>
                    <VAULT_TYPE>Cyber-Ark AIM</VAULT_TYPE>
                    <VAULT_ID>25026922</VAULT_ID>
                    <FOLDER><![CDATA[folder]]></FOLDER>
                    <FILE><![CDATA[file]]></FILE>
                </DIGITAL_VAULT>
            </PRIVATE_KEY_INFO>
            <PASSPHRASE_INFO type="basic">
                <PASSPHRASE><![CDATA[passphrase]]></PASSPHRASE>
            </PASSPHRASE_INFO>
        </PRIVATE_KEY_CERTIFICATE>
        <PRIVATE_KEY_CERTIFICATE>
            <ID>110076922</ID>
            <PRIVATE_KEY_INFO type="basic">
                <PRIVATE_KEY type="rsa">
<![CDATA[-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,F9A653E2D12E019357B349B6EEE068B1

FiLfGHOc0rREmC0cBPsiyqqaitPNYTGeqKRmSBwGNrAzNTAcsKslsoY/WkMDW6QD
dLZNiGB0CFag94zyoMyCjyrdpayACAOWfH5w8VixxHF16Vxx5b6foLBE40FOYAIP
sdm1HvCfSFaN2dPf1Unb0erwjigjJNwYIV78529elE+2+dZIemi90ibh0R35NB60
TLeS3UUVezp/O9ZPLf0pqPPHnWgfW4GXp/SUpwojES9fCQE+BW4MMWHWu8XKtytt
```

```
DcUtGNQlrT205Eg2D/GOwXla//CTHpiP6Zs0pWw/Ohmw1AkPWQa5iGAmCOwqRSFr
...
-----END RSA PRIVATE KEY-----]]></PRIVATE_KEY>
            </PRIVATE_KEY_INFO>
            <PASSPHRASE_INFO type="vault">
                <DIGITAL_VAULT>
                    <VAULT_USERNAME><![CDATA[PASSPHRASE
USERNAME]]></VAULT_USERNAME>
                    <VAULT_TYPE>Quest Vault</VAULT_TYPE>
                    <VAULT_ID>35046922</VAULT_ID>

<SYSTEM_NAME><![CDATA[quest_system_name]]></SYSTEM_NAME>
                </DIGITAL_VAULT>
            </PASSPHRASE_INFO>
            <CERTIFICATE type="openssh">
                <![CDATA[ssh-rsa-cert-v01@openssh.com
AAAAHHNzaC1yc2EtY2VydC12MDFAb3BlbnNzaC5jb20AAAAgwR4bJSiBtJlOgCAQUF3yZ6Io2
WYfnBiOEsQ45RKbqLgAAAADAQABAAABAQC5sVLb7emh8/v2uHp6x1pN5R+MHQwz3A5M3GRKtu
uu1Njc/XYgqeWLMOJpbVtCVXwUcPgKt4Q0DmlGqc4uhZhzrdtpQGHrEivndNNLY9NQj7LozE7
x/sGiWdtmlucUh1teXMaBpM4aER9Y6uW5wv6ZylY7CAV9bcVz/ljlSypmjzkPjJ39AJq+QxZk
Iv+H4uh/T05LwHdilFrjWWwEoI8DV/DRIw3h8o4jhnj1QxBxyjad3efmFaejgRnY6cBW821gm
J3ODQMG96EbWHF6m0vAtmAelx9bahJD8adgu6EF
...
            </CERTIFICATE>
        </PRIVATE_KEY_CERTIFICATE>
    </PRIVATE_KEY_CERTIFICATES>
</UNIX_AUTH_PARAMS>
```

Edit root tools and private-key certificates:

Root tools and private-key certificates in file binary_input_params.xml will be added. Any existing private-key certificates and/or root tools will be deleted from the Unix record.

where edit_params.xml is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<UNIX_AUTH_PARAMS>
    <ROOT_TOOLS>
        <ROOT_TOOL>
            <ID>140016922</ID>
            <STANDARD_TYPE type="pimsu"/>
            <PASSWORD_INFO type="vault">
                <DIGITAL_VAULT>
                    <VAULT_USERNAME><![CDATA[root]]></VAULT_USERNAME>
                    <VAULT_TYPE>Thycotic Secret Server</VAULT_TYPE>
                    <VAULT_ID>25026922</VAULT_ID>

<SECRET_NAME><![CDATA[super_secret_name]]></SECRET_NAME>
```

```
                </DIGITAL_VAULT>
            </PASSWORD_INFO>
        </ROOT_TOOL>
    </ROOT_TOOLS>
    <PRIVATE_KEY_CERTIFICATES>
        <PRIVATE_KEY_CERTIFICATE>
            <ID>110066922</ID>
            <PRIVATE_KEY_INFO type="vault">
                <DIGITAL_VAULT>
                    <VAULT_TYPE>Cyber-Ark AIM</VAULT_TYPE>
                    <VAULT_ID>25026922</VAULT_ID>
                    <FOLDER><![CDATA[folder]]></FOLDER>
                    <FILE><![CDATA[file]]></FILE>
                </DIGITAL_VAULT>
            </PRIVATE_KEY_INFO>
            <PASSPHRASE_INFO type="basic">
                <PASSPHRASE><![CDATA[passphrase]]></PASSPHRASE>
            </PASSPHRASE_INFO>
        </PRIVATE_KEY_CERTIFICATE>
        <PRIVATE_KEY_CERTIFICATE>
            <ID>110076922</ID>
            <PRIVATE_KEY_INFO type="basic">
                <PRIVATE_KEY type="rsa"><![CDATA[-----BEGIN RSA PRIVATE
KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,F9A653E2D12E019357B349B6EEE068B1

FiLfGHOc0rREmC0cBPsiyqqaitPNYTGeqKRmSBwGNrAzNTAcsKslsoY/WkMDW6QD
dLZNiGB0CFag94zyoMyCjyrdpayACAOWfH5w8VixxHF16Vxx5b6foLBE40FOYAIP
sdm1HvCfSFaN2dPf1Unb0erwjigjJNwYIV78529elE+2+dZIemi90ibh0R35NB60
TLeS3UUVezp/O9ZPLf0pqPPHnWgfW4GXp/SUpwojES9fCQE+BW4MMWHWu8XKtytt
...

-----END RSA PRIVATE KEY-----]]></PRIVATE_KEY>
            </PRIVATE_KEY_INFO>
            <PASSPHRASE_INFO type="vault">
                <DIGITAL_VAULT>
                    <VAULT_USERNAME><![CDATA[PASSPHRASE
USERNAME]]></VAULT_USERNAME>
                    <VAULT_TYPE>Quest Vault</VAULT_TYPE>
                    <VAULT_ID>35046922</VAULT_ID>
<SYSTEM_NAME><![CDATA[quest_system_name]]></SYSTEM_NAME>
                </DIGITAL_VAULT>
            </PASSPHRASE_INFO>
            <CERTIFICATE type="openssh">
                <![CDATA[ssh-rsa-cert-v01@openssh.com
AAAAHHNzaC1yc2EtY2VydC12MDFAb3BlbnNzaC5jb20AAAAgwR4bJSiBtJlOgCAQUF3yZ6Io2
WYfnBiOEsQ45RKbqLgAAAADAQABAAABAQC5sVLb7emh8/v2uHp6x1pN5R+MHQwz3A5M3GRKtu
uu1Njc/XYgqeWLMOJpbVtCVXwUcPgKt4Q0DmlGqc4uhZhzrdtpQGHrEi...
```

```
            </CERTIFICATE>
        </PRIVATE_KEY_CERTIFICATE>
    </PRIVATE_KEY_CERTIFICATES>
</UNIX_AUTH_PARAMS>
```

### Delete Root Delegations and Private Keys

Use a Unix record update request including XML binary data like this:

```
curl -H "X-Requested-With: curl" -H "Content-type:text/xml"
-u "USERNAME:PASSWORD" -X "POST"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/action=update&id=12345
67" --data-binary @delete_params.xml
```

Delete all root delegations:

where delete_params.xml is:

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

Delete all private-key certificates:

where delete_params.xml is:

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

Delete all private-key certificates and root delegations:

where delete_params.xml is:

```
<?xml version="1.0" encoding="UTF-8" ?>
<UNIX_AUTH_PARAMS>
    <ROOT_TOOLS></ROOT_TOOLS>
    <PRIVATE_KEY_CERTIFICATES></PRIVATE_KEY_CERTIFICATES>
</UNIX_AUTH_PARAMS>
```

### Unix Record List

API request:

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=list"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_UNIX_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/auth_unix_list_output.
dtd">
<AUTH_UNIX_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2016-09-01T15:59:40Z</DATETIME>
        <AUTH_UNIX_LIST>
            <AUTH_UNIX>
                <ID>1116826922</ID>
                <TITLE>
                    <![CDATA[ssh2]]>
                </TITLE>
                <USERNAME>
                    <![CDATA[root]]>
                </USERNAME>
                <SKIP_PASSWORD>1</SKIP_PASSWORD>
                <ROOT_TOOL_INFO_LIST>
                    <ROOT_TOOL_INFO>
                        <ID>100016922</ID>
                        <ROOT_TOOL>PowerBroker</ROOT_TOOL>
                        <PASSWORD_INFO type="vault">
                            <DIGITAL_VAULT>
                                <DIGITAL_VAULT_ID>
                                    <![CDATA[25026922]]>
                                </DIGITAL_VAULT_ID>
                                <DIGITAL_VAULT_TYPE>
                                    <![CDATA[Cyber-Ark PIM Suite]]>
                                </DIGITAL_VAULT_TYPE>
                                <DIGITAL_VAULT_TITLE>
                                    <![CDATA[CyberArk]]>
                                </DIGITAL_VAULT_TITLE>
                                <VAULT_USERNAME>
                                    <![CDATA[aaa]]>
                                </VAULT_USERNAME>
                                <VAULT_FOLDER>
                                    <![CDATA[aaa]]>
                                </VAULT_FOLDER>
                                <VAULT_FILE>
                                    <![CDATA[bbb]]>
```

```
                    </VAULT_FILE>
                </DIGITAL_VAULT>
            </PASSWORD_INFO>
        </ROOT_TOOL_INFO>
        <ROOT_TOOL_INFO>
            <ID>100006922</ID>
            <ROOT_TOOL>PowerBroker</ROOT_TOOL>
            <PASSWORD_INFO type="basic" />
        </ROOT_TOOL_INFO>
    </ROOT_TOOL_INFO_LIST>
    <PRIVATE_KEY_CERTIFICATE_LIST>
        <PRIVATE_KEY_CERTIFICATE>
            <ID>70016922</ID>
            <PRIVATE_KEY_INFO type="vault">
                <DIGITAL_VAULT>
                    <DIGITAL_VAULT_ID>
                        <![CDATA[25026922]]>
                    </DIGITAL_VAULT_ID>
                    <DIGITAL_VAULT_TYPE>
                        <![CDATA[Cyber-Ark PIM Suite]]>
                    </DIGITAL_VAULT_TYPE>
                    <DIGITAL_VAULT_TITLE>
                        <![CDATA[CyberArk]]>
                    </DIGITAL_VAULT_TITLE>
                    <VAULT_FOLDER>
                        <![CDATA[fff]]>
                    </VAULT_FOLDER>
                    <VAULT_FILE>
                        <![CDATA[gggg]]>
                    </VAULT_FILE>
                </DIGITAL_VAULT>
            </PRIVATE_KEY_INFO>
            <PASSPHRASE_INFO type="basic" />
            <CERTIFICATE type="x.509" />
        </PRIVATE_KEY_CERTIFICATE>
        <PRIVATE_KEY_CERTIFICATE>
            <ID>70006922</ID>
            <PRIVATE_KEY_INFO type="basic">
                <PRIVATE_KEY type="rsa" />
            </PRIVATE_KEY_INFO>
            <PASSPHRASE_INFO type="basic" />
            <CERTIFICATE type="openssh" />
        </PRIVATE_KEY_CERTIFICATE>
    </PRIVATE_KEY_CERTIFICATE_LIST>
    <PORT>22, 23</PORT>
    <IP_SET>
        <IP>10.10.35.253</IP>
    </IP_SET>
    <NETWORK_ID>0</NETWORK_ID>
```

```
                <CREATED>
                    <DATETIME>2016-09-01T09:22:01Z</DATETIME>
                    <BY>quays_as11</BY>
                </CREATED>
                <LAST_MODIFIED>
                    <DATETIME>2016-09-01T15:59:00Z</DATETIME>
                </LAST_MODIFIED>
                <COMMENTS>
                    <![CDATA[vai API cooolio!yay!!!]]>
                </COMMENTS>
                <USE_AGENTLESS_TRACKING>1</USE_AGENTLESS_TRACKING>
                <AGENTLESS_TRACKING_PATH>
                    <![CDATA[/usr/local]]>
                </AGENTLESS_TRACKING_PATH>
            </AUTH_UNIX>
        </AUTH_UNIX_LIST>
    </RESPONSE>
</AUTH_UNIX_LIST_OUTPUT>
```

## Unix Record List Output DTD

Th auth_unix_list_output.dtd has been updated. Changes including new elements for Unix records appear in bold.

Good to Know - These elements will appear in XML output only when the Qualys Shell feature is enabled for your subscription: QUALYS_SHELL, ENABLED, LOG_FACILITY.

```
<!ELEMENT AUTH_UNIX_LIST_OUTPUT (REQUEST?, RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, (AUTH_UNIX_LIST|ID_SET)?, WARNING_LIST?,
GLOSSARY?)>
<!ELEMENT AUTH_UNIX_LIST (AUTH_UNIX+)>

<!ELEMENT AUTH_UNIX (ID, TITLE, USERNAME, SKIP_PASSWORD?,
CLEARTEXT_PASSWORD?, (ROOT_TOOL?|ROOT_TOOL_INFO_LIST?),
((RSA_PRIVATE_KEY?, DSA_PRIVATE_KEY?)|PRIVATE_KEY_CERTIFICATE_LIST?),
```

```
PORT?, IP_SET, LOGIN_TYPE?, DIGITAL_VAULT?, NETWORK_ID?, CREATED,
LAST_MODIFIED, COMMENTS?, USE_AGENTLESS_TRACKING?,
AGENTLESS_TRACKING_PATH?, QUALYS_SHELL?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT SKIP_PASSWORD (#PCDATA)>
<!ELEMENT CLEARTEXT_PASSWORD (#PCDATA)>
<!ELEMENT ROOT_TOOL (#PCDATA)>
<!ELEMENT ROOT_TOOL_INFO_LIST (ROOT_TOOL_INFO)*>
<!ELEMENT RSA_PRIVATE_KEY EMPTY>
<!ELEMENT DSA_PRIVATE_KEY EMPTY>
<!ELEMENT PRIVATE_KEY_CERTIFICATE_LIST (PRIVATE_KEY_CERTIFICATE)*>
<!ELEMENT PORT (#PCDATA)>

<!ELEMENT IP_SET (IP|IP_RANGE)+>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>
<!ELEMENT LOGIN_TYPE (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT CREATED (DATETIME, BY)>
<!ELEMENT BY (#PCDATA)>
<!ELEMENT LAST_MODIFIED (DATETIME)>
<!ELEMENT COMMENTS (#PCDATA)>
<!ELEMENT USE_AGENTLESS_TRACKING (#PCDATA)>
<!ELEMENT AGENTLESS_TRACKING_PATH (#PCDATA)>
<!ELEMENT QUALYS_SHELL (ENABLED, LOG_FACILITY?)>

<!ELEMENT ROOT_TOOL_INFO (ID, ROOT_TOOL, PASSWORD_INFO?)>
<!ELEMENT PASSWORD_INFO (DIGITAL_VAULT?)>
<!ATTLIST PASSWORD_INFO type (basic|vault) "basic">

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

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

<!ELEMENT CERTIFICATE EMPTY>
<!ATTLIST CERTIFICATE type (x.509|openssh) #REQUIRED>

<!ELEMENT DIGITAL_VAULT (DIGITAL_VAULT_ID, DIGITAL_VAULT_TYPE,
DIGITAL_VAULT_TITLE, VAULT_USERNAME?, VAULT_FOLDER?, VAULT_FILE?,
```

```
VAULT_SECRET_NAME?, VAULT_SYSTEM_NAME?, VAULT_EP_NAME?, VAULT_EP_TYPE?,
VAULT_EP_CONT?, VAULT_NS_TYPE?, VAULT_NS_NAME?)>
<!ELEMENT DIGITAL_VAULT_ID (#PCDATA)>
<!ELEMENT DIGITAL_VAULT_TYPE (#PCDATA)>
<!ELEMENT DIGITAL_VAULT_TITLE (#PCDATA)>
<!ELEMENT VAULT_USERNAME (#PCDATA)>
<!ELEMENT VAULT_FOLDER (#PCDATA)>
<!ELEMENT VAULT_FILE (#PCDATA)>
<!ELEMENT VAULT_SECRET_NAME (#PCDATA)>
<!ELEMENT VAULT_SYSTEM_NAME (#PCDATA)>
<!ELEMENT VAULT_EP_NAME (#PCDATA)>
<!ELEMENT VAULT_EP_TYPE (#PCDATA)>
<!ELEMENT VAULT_EP_CONT (#PCDATA)>
<!ELEMENT VAULT_NS_TYPE (#PCDATA)>
<!ELEMENT VAULT_NS_NAME (#PCDATA)>

<!ELEMENT ENABLED (#PCDATA)>
<!ELEMENT LOG_FACILITY (#PCDATA)>

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

<!ELEMENT GLOSSARY (USER_LIST?)>
<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>
<!-- EOF -->
```

# New Support for Cyber-Ark AIM Vaults

This new vault type can be used to retrieve authentication credentials from CyberArk's Central Credential Provider (CCP) solution. We updated the authentication record API (create, update, list) and the vault API (create, update, list, view) to support the new vault type. There are new input parameters for create/update vault.

## Authentication Vault API

DTD update:

We added the APPID element.

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

...
<!ELEMENT RESPONSE (DATETIME, VAULT_QUEST)>
<!ELEMENT VAULT_QUEST (TITLE, COMMENTS, VAULT_TYPE, CREATED_ON?, OWNER?,
LAST_MODIFIED?, APPID?, USERNAME?, URL?, SSL_VERIFY?, DOMAIN?,
API_USERNAME?, WEB_USERNAME?, SERVER_ADDRESS?, PORT?, SAFE?, (UUID|ID))>
<!ELEMENT UUID (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT COMMENTS (#PCDATA)>
<!ELEMENT VAULT_TYPE (#PCDATA)>
<!ELEMENT CREATED_ON (#PCDATA)>
<!ELEMENT OWNER (#PCDATA)>
<!ELEMENT APPID (#PCDATA)>
<!ELEMENT USERNAME (#PCDATA)>
...
<!ELEMENT LAST_MODIFIED (DATETIME, BY?)>
<!ELEMENT BY (#PCDATA)>
<!-- EOF -->
```

### List Authentication Vault

Use the parameter “action=list” to list the vaults defined in your account. To view a specific Cyber-Ark AIM vault, specify the ID or title of the vault.

API request (Windows):

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=list&title=New-CyberArk-AIM"
"https://qualysapi.qualys.com/api/2.0/fo/vault/"
```

XML output (Windows):

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

```
<!DOCTYPE AUTH_VAULT_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/vault/vault_output.dtd">
<AUTH_VAULT_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2016-09-08T06:37:08Z</DATETIME>
    <STATUS>Success</STATUS>
    <COUNT>1</COUNT>
    <AUTH_VAULTS>
      <AUTH_VAULT>
        <TITLE><![CDATA[New-CyberArk-AIM]]></TITLE>
        <VAULT_TYPE><![CDATA[Cyber-Ark AIM]]></VAULT_TYPE>
        <LAST_MODIFIED>
          <DATETIME>2016-09-07T11:45:29Z</DATETIME>
          <BY>user_john</BY>
        </LAST_MODIFIED>
        <ID>7004</ID>
      </AUTH_VAULT>
    </AUTH_VAULTS>
  </RESPONSE>
</AUTH_VAULT_LIST_OUTPUT>
```

### Create Cyber-Ark AIM Authentication Vault

Use the parameter “action=create” to create a Cyber-Ark AIM authentication vault in your account.

| Parameter | Description |
|---|---|
| action=create | (Required) |
| type={value} | (Required) The vault type. A valid value is:<br>Cyber-Ark PIM Suite<br>Thycotic Secret Server<br>Quest Vault<br>CA Access Control<br>Hitachi ID PAM<br>Lieberman ERPM<br>Cyber-Ark AIM |
| appid={value} | (Required) Application ID string defined by the customer. The application ID acts as an authenticator for our scanner to call CCP web services API. The maximum length of an application ID name is 128 bytes and the first 28 characters must be unique (leading and/or trailing space or periods in the input value will be removed). These restricted words cannot be included in a application ID: Users, Addresses, Areas, XUserRules, unknown, Locations, Safes, Schedule, VaultCategories, Builtin. These special characters cannot be included in a application ID: \ / : * ? " < > \| \t \r \n \x1F. |

| Parameter | Description |
|---|---|
| safe={value} | (Required) The name of the digital password safe. The safe name can contain a maximum of 28 characters (leading and/or trailing space in the input value will be removed). These special characters cannot be included in a safe name: \ / : * ? " < > \| \t \r \n \x1F |
| url={value} | (Required) The HTTP or HTTPS URL over SSL protocols to access CyberArk's CCP web services. |
| ssl_verify={1\|0} | (Required) When set to 1, our service will verify the CCP SSL certificate of the web server to make sure the certificate is valid and trusted. When set to 0 our service will not verify the certificate of the web server. |
| certificate={value} | (Optional) You must include an X.509 certificate with your private key. Enter the certificate block after the key block and be sure to include the first and last line (-----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----). |
| | For a create/update request, if the certificate parameter is specified, then the private_key parameter must also be specified. |
| private_key={value} | (Optional) Specify private key for authentication. Copy the contents of private key file (id_rsa) and be sure to include the first and last line (-----BEGIN PRIVATE KEY----- and -----END PRIVATE KEY-----). |
| | For a create/update request, if the private_key parameter is specified, then the certificate parameter must also be specified. |
| password={value} | (Optional) Specify a password for the encrypted private_key. |

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=create&type=Cyber-Ark AIM&title=New-CyberArk-
AIM&appid=CyberArk007&safe=Vaultsafe&url=https://afco.com&ssl_verify=1&
cert=-----BEGIN+CERTIFICATE-----
%0D%0AMIIDXzCCAkcCAQEwDQYJKoZIwdjELMAkGA1UEBhM%0D%0A-----END+CERTIFICATE
-----&private_key_pwd=password&private_key=-----BEGIN+RSA+PRIVATE+KEY----
-%0D%0AMIIEowIBAAKCAQEAmbSGAPwS662q5SsJ2XA2mVvKOfXa%2%0D%0A-----
END+RSA+PRIVATE+KEY-----"
"https://qualysapi.qualys.com/api/2.0/fo/vault/index.php"
```

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>2016-09-02T06:10:02Z</DATETIME>
    <TEXT>Success</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>7004</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

#### Update Cyber-Ark AIM Authentication Vault

Use the parameter "action=update" to update a Cyber-Ark AIM authentication vault in your account.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=update&id=7004&appid=appID-
update&safe=safeupdate&url=http://afco.com&ssl_verify=0&private_key_pwd=
passwordupdate"
"https://qualysapi.qualys.com/api/2.0/fo/vault/index.php"
```

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>2016-09-02T06:10:02Z</DATETIME>
    <TEXT>Success</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>11527913</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

#### View Cyber-Ark AIM Authentication Vault

Use the parameter "action=view" and specify the ID of the Cyber-Ark AIM authentication vault to view its details.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=view&id=7004"
"https://qualysapi.qualys.com/api/2.0/fo/vault/index.php"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE VAULT_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/vault/vault_view.dtd">
<VAULT_OUTPUT>
  <RESPONSE>
    <DATETIME>2016-09-08T06:38:28Z</DATETIME>
    <VAULT_QUEST>
      <TITLE><![CDATA[New Cyber-Ark AIM Vault]]></TITLE>
      <COMMENTS><![CDATA[]]></COMMENTS>
      <VAULT_TYPE><![CDATA[Cyber-Ark AIM]]></VAULT_TYPE>
      <CREATED_ON>2016-09-07T07:09:34Z</CREATED_ON>
      <OWNER>user_john</OWNER>
      <LAST_MODIFIED>
        <DATETIME>2016-09-08T06:37:49Z</DATETIME>
        <BY>user_john</BY>
      </LAST_MODIFIED>
      <APPID><![CDATA[735435]]></APPID>
      <URL><![CDATA[https://afco.com]]></URL>
      <SSL_VERIFY><![CDATA[1]]></SSL_VERIFY>
      <SAFE><![CDATA[56908456904]]></SAFE>
      <ID>7004</ID>
    </VAULT_QUEST>
  </RESPONSE>
</VAULT_OUTPUT>
```

## Authentication Record API

You can now list, create/update authentication record for Cyber-Ark AIM vaults.

#### List Authentication Record

API request (Windows):

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=list&details=All&ids=11176"
"https://qualysapi.qualys.com/api/2.0/fo/auth/windows/"
```

XML output (Windows):

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_WINDOWS_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/windows/auth_windows_list_
```

```
output.dtd">
<AUTH_WINDOWS_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2016-09-08T06:57:19Z</DATETIME>
    <AUTH_WINDOWS_LIST>
      <AUTH_WINDOWS>
        <ID>11176</ID>
        <TITLE><![CDATA[create_auth_api]]></TITLE>
        <USERNAME><![CDATA[user_john]]></USERNAME>
        <NTLM_V2>1</NTLM_V2>
        <CLEARTEXT_PASSWORD>0</CLEARTEXT_PASSWORD>
        <IP_SET>
          <IP>10.10.10.28</IP>
        </IP_SET>
        <LOGIN_TYPE><![CDATA[vault]]></LOGIN_TYPE>
        <DIGITAL_VAULT>
          <DIGITAL_VAULT_ID><![CDATA[7004]]></DIGITAL_VAULT_ID>
          <DIGITAL_VAULT_TYPE><![CDATA[Cyber-Ark AIM]]>
          </DIGITAL_VAULT_TYPE>
          <DIGITAL_VAULT_TITLE><![CDATA[Cyber-Ark AIM 007]]>
          </DIGITAL_VAULT_TITLE>
          <VAULT_FOLDER><![CDATA[vaults\new_vault]]></VAULT_FOLDER>
          <VAULT_FILE><![CDATA[myfl.txt]]></VAULT_FILE>
        </DIGITAL_VAULT>
        <CREATED>
          <DATETIME>2016-09-07T05:17:19Z</DATETIME>
          <BY>user_john</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2016-09-08T06:53:39Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_WINDOWS>
    </AUTH_WINDOWS_LIST>
    <GLOSSARY>
      <USER_LIST>
        <USER>
          <USER_LOGIN>user_john</USER_LOGIN>
          <FIRST_NAME>john</FIRST_NAME>
          <LAST_NAME>doe</LAST_NAME>
        </USER>
      </USER_LIST>
    </GLOSSARY>
  </RESPONSE>
</AUTH_WINDOWS_LIST_OUTPUT>
```

### Create Authentication Record

API request (Windows):

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=create&title=API_v2_utwrx_mp_Windows_3&username=Qualys&ips=10.10.
10.28&login_type=vault&vault_id=6040&vault_type=Cyber-Ark
AIM&file=vtfile&folder=vaults\new_vault"
"https://qualysapi.qualys.com/api/2.0/fo/auth/windows/"
```

XML output (Windows):

```
<?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>2016-08-30T11:34:58Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>31407913</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

### Update Authentication Record

API request (Windows):

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=update&ids=31407913&login_type=vault&vault_id=7004&vault_type=Cyb
er-Ark AIM&folder=www\tem&file=myfl.txt&title=ceatauthapi_dp&
username=admin&ips=10.10.10.28"
"https://qualysapi.qualys.com/api/2.0/fo/auth/windows/"
```

XML output (Windows):

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qqualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2016-09-08T06:53:39Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Updated</TEXT>
        <ID_SET>
          <ID>31407913</ID>
```

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

# Launch Scan using All Scanners in Network

You can now launch and schedule scans using the All Scanners in Network option, which will launch scans using all the scanner appliances in your network. Simply specify the network you want to scan and the new input parameter “scanners_in_network=1”.

Applies to vulnerability scans, compliance scans, and scheduled VM scans, when the Network Support feature is enabled for your subscription.

## VM Scan API (v2)

API request:

```
curl -u 'username:password' -H 'X-Requested-With:curl demo' -d
"action=launch&scan_title=scan3&option_title=Initial+Options&ip_network_i
d=12807913&scanners_in_network=1&asset_groups=AG1-GDN"
"https://qualysapi.qualys.com/api/2.0/fo/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>2016-09-15T21:53:45Z</DATETIME>
    <TEXT>New vm scan launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>18197</VALUE>
      </ITEM>
      <ITEM>
        <KEY>REFERENCE</KEY>
        <VALUE>scan/1473976424.18197</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

## VM Scan Schedule API (v2)

```
curl -u 'username:password' -H 'X-Requested-With:curl demo 2' -d
"action=create&scan_title=API+Schedule+scan&option_title=Initial+Options&
ip_network_id=12807913&scanners_in_network=1&ip=10.10.10.10,10.10.10.11&o
ccurrence=monthly&frequency_months=12&day_of_month=20&start_minute=00&sta
rt_hour=22&time_zone_code=IN&observe_dst=no&pause_after_hours=3&resume_in
_days=4&recurrence=5&start_date=08/20/2016&active=1"
```

```
"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>2016-09-15T21:57:25Z</DATETIME>
    <TEXT>New scan scheduled successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>13349</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

## PC Scan API (v2)

API request:

```
curl -u 'username:password' -H 'X-Requested-With:curl demo 2' -d
"action=launch&scan_title=pc+scan+API&option_id=3262&ip_network_id=128079
13&scanners_in_network=1&ip=10.10.10.10,10.10.10.11"
"https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/"
```

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>2016-09-15T21:55:36Z</DATETIME>
    <TEXT>New compliance scan launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>18198</VALUE>
      </ITEM>
      <ITEM>
        <KEY>REFERENCE</KEY>
        <VALUE>compliance/1473976536.18198</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

# Appliance API - Add tags to your scanner appliances

You can now add tags to your scanner appliances using the Appliance API v2 (/api/2.0/fo/appliance). The new parameters let you add, remove and reset tags for appliances.

| Parameter | Description |
|---|---|
| action=update | (Required) |
| set_tags= {value} | (Optional) Specify tag to be assigned to the scanner appliance. Both virtual and physical scanners can be tagged. |
| | These parameters are mutually exclusive and cannot be specified in the same request: set_tags and add_tags, remove_tags. |
| add_tags= {value} | (Optional) Specify tag to be added to the existing list of tags assigned to the scanner. Multiple entries are comma separated. |
| | These parameters are mutually exclusive and cannot be specified in the same request: set_tags and add_tags, remove_tags. |
| remove_tags= {value} | (Optional) Specify tag to be removed from the existing list of tags assigned to scanner. Multiple entries are comma separated. |
| | These parameters are mutually exclusive and cannot be specified in the same request: set_tags and add_tags, remove_tags. |
| tag_set_by= {id \| name} | (Optional) Specify “id” (the default) to select a tag set by providing tag IDs. Specify “name” to select a tag set by providing tag names. |

### Sample 1

The request will add tags for windows agent and remove tags for linux agents.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -X POST -d
"action=update&id=3105&tag_set_by=name&add_tags=windows_agent&remove_tags
=linux_agents"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

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>2016-09-15T19:44:35Z</DATETIME>
    <TEXT>Virtual scanner updated successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>3105</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

#### Sample 2

This request will assign the tags local_host and local_IP to the scanner appliance.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -X POST -d
"action=update&id=3112&tag_set_by=name&set_tags=local_host,local_IP"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

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>2016-09-15T19:47:37Z</DATETIME>
    <TEXT>Virtual scanner updated successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>3112</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
```

# Physical Scanner Appliance API to update VLAN and Static Routes

Using the new Physical Scanner Appliance API v2 (/api/2.0/fo/appliance/physical/ ), Managers and Unit Managers can update physical scanner appliances.

## Update Physical Scanner

Use these parameters:

| Parameter | Description |
|---|---|
| action=update | (Required) The POST method must be used. |
| id={id} | (Required) A valid ID of a physical scanner. |
| name={string} | (Optional) The friendly name. This name can't already be assigned to an appliance in your account. It can be a maximum of 15 characters, spaces are not allowed. |
| polling_interval={value} | (Optional) The polling interval, in seconds. A valid value is 60 to 3600 (we recommend 180 which is the default). This is the frequency that the physical scanner will attempt to connect to our Cloud Security Platform. The appliance calls home to provide health updates/heartbeats to the platform, to get software updates from the platform, to learn if new scan jobs have been requested by users, and to upload scan results data to the platform, if applicable. |
| set_vlans={value} | Use this parameter to specify one or more VLANs for scanner. |
| set_tags= {value} | (Optional) Specify tag to be assigned to the scanner appliance. Both virtual and physical scanners can be tagged. |
| | These parameters are mutually exclusive and cannot be specified in the same request: set_tags and add_tags, remove_tags. |
| add_tags= {value} | (Optional) Specify tag to be added to the existing list of tags assigned to the scanner. Multiple entries are comma separated. |
| | These parameters are mutually exclusive and cannot be specified in the same request: set_tags and add_tags, remove_tags. |

| Parameter | Description |
|---|---|
| remove_tags= {value} | (Optional) Specify tag to be removed from the existing list of tags assigned to scanner. Multiple entries are comma separated.<br><br>These parameters are mutually exclusive and cannot be specified in the same request: set_tags and add_tags, remove_tags. |
| tag_set_by= {id\|name} | (Optional) Specify "id" (the default) to select a tag set by providing tag IDs. Specify "name" to select a tag set by providing tag names. |
| set_routes={value} | Use this parameter to specify one or more routes for scanner. |
| comment={value} | (Optional) User-defined comments. |

### Sample 1

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=update&id=5115&comment=Hello"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/physical/"
```

### Sample 2

Add VLAN and routes with Name, Polling interval and comments to Physical scanner:

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X POST -d
"action=update&id=5115&name=physcanner&polling_interval=360&set_routes=10
.10.10.10|255.255.255.0|10.10.10.10|routes1&set_vlans=1|10.2.0.2|255.255.
255.0|Testvlan1&comment=Update_scanner"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/physical/"
```

### Sample 3

Update physical scanner using tag_set_by and add_tags parameters:

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=update&id=5115&tag_set_by=id&add_tags=7691422"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/physical/"
```

#### Sample 4

Update physical scanner using tag_set_by and set_tags parameters:

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=update&id=5115&tag_set_by=id&set_tags=7691422"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/physical/"
```

#### Sample 5

Update physical scanner using tag_set_by and remove_tags parameters:

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=update&id=5115&tag_set_by=id&remove_tags=7691422"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/physical/"
```

XML output:

The XML output uses the simple return (/api/2.0/simple_return.dtd).

```
<?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>2016-10-01T00:12:29Z</DATETIME>
        <TEXT>Physical scanner updated successfully</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>5115</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

# Appliance List Output - Start date/time for CMD Only mode added

The Appliance List Output now includes the date/time an appliance enters into CMD Only (command only) mode. This mode may be entered for various reasons, such as when a session expires.

API request:

```
curl -H "X-Requested-With: curl demo" -u USERNAME:PASSWORD -d
"action=list&output_mode=full"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

XML output:

```
<APPLIANCE_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2016-09-08T06:17:49Z</DATETIME>
    <APPLIANCE_LIST>
      <APPLIANCE>
        <ID>7108</ID>
        <UUID>018a4f9f-a839-565f-8391-78606155aca1</UUID>
        <NAME>my_vs_1</NAME>
        <SOFTWARE_VERSION>2.6</SOFTWARE_VERSION>
        <RUNNING_SLICES_COUNT>0</RUNNING_SLICES_COUNT>
        <RUNNING_SCAN_COUNT>0</RUNNING_SCAN_COUNT>
        <STATUS>Online</STATUS>
        <CMD_ONLY_START>2016-09-07 03:21:03</CMD_ONLY_START>
...
```

DTD update:

We added the CMD_ONLY_START element to the Appliance List Output DTD (appliance_list_output.dtd).

```
<!-- QUALYS APPLIANCE_LIST_OUTPUT DTD -->
...
<!ELEMENT APPLIANCE (ID, UUID, NAME, NETWORK_ID?, SOFTWARE_VERSION,
                     RUNNING_SLICES_COUNT, RUNNING_SCAN_COUNT, STATUS,
                     CMD_ONLY_START?, MODEL_NUMBER?, SERIAL_NUMBER?,
                     ACTIVATION_CODE?, INTERFACE_SETTINGS*,
                     PROXY_SETTINGS?, 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 ID (#PCDATA)>
<!ELEMENT UUID (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT SOFTWARE_VERSION (#PCDATA)>
<!ELEMENT RUNNING_SLICES_COUNT (#PCDATA)>
<!ELEMENT RUNNING_SCAN_COUNT (#PCDATA)>
<!ELEMENT STATUS (#PCDATA)>
<!ELEMENT CMD_ONLY_START (#PCDATA)>
...
```

# User List Output - User ID added

The User List v1 API (/msp/user_list.php) lets you view the users in the subscription. The user list output now includes the user ID assigned to each user.

DTD update:

The USER_ID element is added to the user list output DTD (user_list_output.dtd).

```
<!-- QUALYS USER LIST OUTPUT DTD -->
<!ELEMENT USER_LIST_OUTPUT (ERROR | USER_LIST)>

<!ELEMENT ERROR (#PCDATA)*>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!ELEMENT USER_LIST (USER*)>

<!ELEMENT USER (USER_LOGIN?, USER_ID?, EXTERNAL_ID?, CONTACT_INFO,
ASSIGNED_ASSET_GROUPS?, USER_STATUS, CREATION_DATE, LAST_LOGIN_DATE?,
USER_ROLE?, BUSINESS_UNIT?, UNIT_MANAGER_POC?, MANAGER_POC?,
UI_INTERFACE_STYLE?, PERMISSIONS?, NOTIFICATIONS?)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT USER_ID (#PCDATA)>

<!ELEMENT EXTERNAL_ID (#PCDATA)>

...

<!ELEMENT MAP (#PCDATA)>
<!ELEMENT SCAN (#PCDATA)>
<!ELEMENT DAILY_TICKETS (#PCDATA)>
```

### Sample Request

API request:

```
curl -u "USERNAME:PASSWORD" "X-Requested-With: curl"
"https://qualysapi.qualys.com/msp/user_list.php"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE USER_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/user_list_output.dtd">
<USER_LIST_OUTPUT>
  <USER_LIST>
    <USER>
      <USER_LOGIN>john_user</USER_LOGIN>
      <USER_ID>1057</USER_ID>
      <CONTACT_INFO>
```

```
            <FIRSTNAME><![CDATA[John]]></FIRSTNAME>
            <LASTNAME><![CDATA[Doe]]></LASTNAME>
            <TITLE><![CDATA[API]]></TITLE>
            <PHONE><![CDATA[978123456]]></PHONE>
            <FAX><![CDATA[]]></FAX>
            <EMAIL><![CDATA[jdoe@afco.com]]></EMAIL>
            <COMPANY><![Afco[Network]]></COMPANY>
            <ADDRESS1><![CDATA[500 First Street]]></ADDRESS1>
            <ADDRESS2><![CDATA[]]></ADDRESS2>
            <CITY><![CDATA[Jersey City]]></CITY>
            <COUNTRY>United States of America</COUNTRY>
            <STATE>New Jersey</STATE>
            <ZIP_CODE><![CDATA[07303]]></ZIP_CODE>
            <TIME_ZONE_CODE><![CDATA[Auto]]></TIME_ZONE_CODE>
          </CONTACT_INFO>
          <USER_STATUS>Active</USER_STATUS>
          <CREATION_DATE>2016-08-01T06:42:04Z</CREATION_DATE>
          <LAST_LOGIN_DATE>2016-08-30T11:45:00Z</LAST_LOGIN_DATE>
          <USER_ROLE>Manager</USER_ROLE>
          <BUSINESS_UNIT><![CDATA[Manufacturing]]></BUSINESS_UNIT>
          <UNIT_MANAGER_POC>0</UNIT_MANAGER_POC>
          <MANAGER_POC>1</MANAGER_POC>
          <UI_INTERFACE_STYLE>standard_blue</UI_INTERFACE_STYLE>
          <PERMISSIONS>
            <CREATE_OPTION_PROFILES>1</CREATE_OPTION_PROFILES>
            <PURGE_INFO>1</PURGE_INFO>
            <ADD_ASSETS>1</ADD_ASSETS>
            <EDIT_REMEDIATION_POLICY>1</EDIT_REMEDIATION_POLICY>
            <EDIT_AUTH_RECORDS>1</EDIT_AUTH_RECORDS>
          </PERMISSIONS>
          <NOTIFICATIONS>
            <LATEST_VULN>weekly</LATEST_VULN>
            <MAP>ags</MAP>
            <SCAN>ags</SCAN>
            <DAILY_TICKETS>0</DAILY_TICKETS>
          </NOTIFICATIONS>
        </USER>
      </USER_LIST>
    </USER_LIST_OUTPUT>
```

# MS SQL Authentication Record API - Domain supported

Now you can easily create domain based MS SQL authentication records. Just add the member domain to your MS SQL record and we'll auto discover MS SQL instances for authentication.

DTD update:

We added the MEMBER_DOMAIN element (auth_ms_sql_list_output.dtd).

```
<!-- QUALYS AUTH_MS_SQL_LIST_OUTPUT DTD -->
<!ELEMENT AUTH_MS_SQL_LIST_OUTPUT (REQUEST?, RESPONSE)>
...
<!ELEMENT AUTH_MS_SQL (ID, TITLE, USERNAME, NTLM_V1?, NTLM_V2?, KERBEROS?,
(INSTANCE | AUTO_DISCOVER_INSTANCES), (DATABASE |
AUTO_DISCOVER_DATABASES), (PORT|AUTO_DISCOVER_PORTS), DB_LOCAL,
WINDOWS_DOMAIN?, (IP_SET|MEMBER_DOMAIN), NETWORK_ID?, CREATED,
LAST_MODIFIED, COMMENTS?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT NTLM_V1 (#PCDATA)>
<!ELEMENT NTLM_V2 (#PCDATA)>
<!ELEMENT KERBEROS (#PCDATA)>
<!ELEMENT INSTANCE (#PCDATA)>
<!ELEMENT DATABASE (#PCDATA)>
<!ELEMENT PORT (#PCDATA)>
<!ELEMENT DB_LOCAL (#PCDATA)>
<!ELEMENT WINDOWS_DOMAIN (#PCDATA)>
<!ELEMENT AUTO_DISCOVER_INSTANCES (#PCDATA)>
<!ELEMENT AUTO_DISCOVER_DATABASES (#PCDATA)>
<!ELEMENT AUTO_DISCOVER_PORTS (#PCDATA)>

<!ELEMENT IP_SET (IP|IP_RANGE)+>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>
<!ELEMENT MEMBER_DOMAIN (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT CREATED (DATETIME, BY)>
...
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>

<!-- EOF -->
```

## MS SQL Record: List

Use the parameter “action=list” to list the MS SQL servers defined in your domain. To view a specific MS SQL record, specify the ID of the authentication record.

API request (Windows):

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=list&echo_request=1&ids=13907"
"https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/"
```

XML output (Windows):

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_MS_SQL_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/auth_ms_sql_list_out
put.dtd">
<AUTH_MS_SQL_LIST_OUTPUT>
  <REQUEST>
    <DATETIME>2016-09-20T05:34:37Z</DATETIME>
    <USER_LOGIN>user_john</USER_LOGIN>
    <RESOURCE>
       https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/</RESOURCE>
    <PARAM_LIST>
      <PARAM>
        <KEY>action</KEY>
        <VALUE>list</VALUE>
      </PARAM>
      <PARAM>
        <KEY>echo_request</KEY>
        <VALUE>1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>ids</KEY>
        <VALUE>13907</VALUE>
      </PARAM>
    </PARAM_LIST>
  </REQUEST>
  <RESPONSE>
    <DATETIME>2016-09-20T05:34:37Z</DATETIME>
    <AUTH_MS_SQL_LIST>
      <AUTH_MS_SQL>
        <ID>13907</ID>
        <TITLE><![CDATA[mssqlvt4]]></TITLE>
        <USERNAME><![CDATA[administrator]]></USERNAME>
        <NTLM_V2>1</NTLM_V2>
        <KERBEROS>1</KERBEROS>
        <INSTANCE><![CDATA[MSSQLSERVER]]></INSTANCE>
        <DATABASE><![CDATA[master]]></DATABASE>
        <PORT>8012</PORT>
```

```
        <DB_LOCAL>1</DB_LOCAL>
        <MEMBER_DOMAIN><![CDATA[sitedomain.com]]></MEMBER_DOMAIN>
        <NETWORK_ID>0</NETWORK_ID>
        <CREATED>
          <DATETIME>2016-09-20T05:26:31Z</DATETIME>
          <BY>user_john</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2016-09-20T05:26:31Z</DATETIME>
        </LAST_MODIFIED>
        <COMMENTS><![CDATA[authcreated]]></COMMENTS>
      </AUTH_MS_SQL>
    </AUTH_MS_SQL_LIST>
  </RESPONSE>
</AUTH_MS_SQL_LIST_OUTPUT>
```

## MS SQL Record: Create

To create a MS SQL authentication record, you must either specify the member domain or ips parameter.

| Parameter | Description |
|---|---|
| action=create/update | (Required) |
| member_domain={value} | (Optional) Defines the domain of the MS SQL server for the authentication record. |
| | For create request, it is required to specify either ips or this parameter. |
| | This parameter and the ips parameter cannot be specified in the same request. |

API request (Windows):

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=create&title=mssqlvt1&username=administrator&password=abc123&db_l
ocal=1&port=8012&member_domain=sitedomain.com&echo_request=1&comments=aut
hcreated&instance=MSSQLSERVER&database=master"
"https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/"
```

XML output (Windows):

```
 <?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>2016-09-20T05:26:31Z</DATETIME>
    <USER_LOGIN>user_john</USER_LOGIN>
```

```
<RESOURCE>
   https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/</RESOURCE>
<PARAM_LIST>
  <PARAM>
    <KEY>action</KEY>
    <VALUE>create</VALUE>
  </PARAM>
  <PARAM>
    <KEY>title</KEY>
    <VALUE>mssqlvt4</VALUE>
  </PARAM>
  <PARAM>
    <KEY>username</KEY>
    <VALUE>administrator</VALUE>
  </PARAM>
  <PARAM>
    <KEY>password</KEY>
    <VALUE>abc123</VALUE>
  </PARAM>
  <PARAM>
    <KEY>db_local</KEY>
    <VALUE>1</VALUE>
  </PARAM>
  <PARAM>
    <KEY>port</KEY>
    <VALUE>8012</VALUE>
  </PARAM>
  <PARAM>
    <KEY>member_domain</KEY>
    <VALUE>sitedomain.com</VALUE>
  </PARAM>
  <PARAM>
    <KEY>echo_request</KEY>
    <VALUE>1</VALUE>
  </PARAM>
  <PARAM>
    <KEY>comments</KEY>
    <VALUE>authcreated</VALUE>
  </PARAM>
  <PARAM>
    <KEY>instance</KEY>
    <VALUE>MSSQLSERVER</VALUE>
  </PARAM>
  <PARAM>
    <KEY>database</KEY>
    <VALUE>master</VALUE>
  </PARAM>
 </PARAM_LIST>
</REQUEST>
```

```
  <RESPONSE>
    <DATETIME>2016-09-20T05:26:31Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>13907</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

## MS SQL Record: Update

Use the parameter "action=update" to update MS SQL authentication record.

API request (Windows):

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=update&echo_request=1&ids=13907&member_domain=webdomain.com"
"https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/"
```

XML output (Windows):

```
<?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>2016-09-20T05:37:13Z</DATETIME>
    <USER_LOGIN>user_john</USER_LOGIN>
    <RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql/
      </RESOURCE>
    <PARAM_LIST>
      <PARAM>
        <KEY>action</KEY>
        <VALUE>update</VALUE>
      </PARAM>
      <PARAM>
        <KEY>echo_request</KEY>
        <VALUE>1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>ids</KEY>
        <VALUE>13907</VALUE>
      </PARAM>
      <PARAM>
        <KEY>member_domain</KEY>
        <VALUE>webdomain.com</VALUE>
```

```
        </PARAM>
      </PARAM_LIST>
    </REQUEST>
    <RESPONSE>
      <DATETIME>2016-09-20T05:37:13Z</DATETIME>
      <BATCH_LIST>
        <BATCH>
          <TEXT>Successfully Updated</TEXT>
          <ID_SET><ID>13907</ID>
           </ID_SET>
        </BATCH>
      </BATCH_LIST>
    </RESPONSE>
  </BATCH_RETURN>
```

# IP Update - Fix to Command List Output and DTD

The Command List Output DTD is used when you perform an IP update that results in a warning about duplicate hosts. We made a fix in the XML output to add the opening tag for COMMAND_LIST_OUTPUT, and we updated the COMMAND_LIST_OUTPUT DTD to include missing elements CODE and WARNING (plus sub-elements).

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With:curl demo2" -X POST
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/?action=update&tracking
_method=NETBIOS&ips=10.10.10.28"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE COMMAND_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/appliance/command/command_list_o
utput.dtd">
<COMMAND_LIST_OUTPUT>
  <RESPONSE>
    <CODE>1982</CODE>
    <DATETIME>2016-08-18T04:48:52Z</DATETIME>
    <WARNING>
      <TEXT>You cannot change the tracking method for the following host
using the API since there are multiple scan data entries. This can happen
when the host is resolved to different hostnames in different scan tasks.
You'll need to change the tracking method using the UI. Use the URL to log
into your account, edit the host and select another tracking method. At
the prompt click Apply to save the most recent scan data and purge the
other scan data.</TEXT>
      <DUPLICATE_HOSTS>
        <DUPLICATE_HOST>
          <IP>10.10.10.28</IP>
          <DNS_HOSTNAME>vistasp1-26-107</DNS_HOSTNAME>
          <NETBIOS_HOSTNAME>VISTASP1-26-107</NETBIOS_HOSTNAME>
          <LAST_SCANDATE>02/29/2016 at 08:53:34 (GMT)</LAST_SCANDATE>
          <TRACKING>IP</TRACKING>
        </DUPLICATE_HOST>
      </DUPLICATE_HOSTS>
<URL><![CDATA[https://qualysapi.qualys.com/fo/tools/ip_assets.php]]></URL
>
    </WARNING>
  </RESPONSE>
</COMMAND_LIST_OUTPUT>
```

DTD update:

```
<!-- QUALYS COMMAND_LIST_OUTPUT DTD -->
```

```
<!-- $Revision$ -->
<!ELEMENT COMMAND_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 (CODE?, DATETIME, COMMAND_LIST?, WARNING?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT COMMAND_LIST (COMMAND+|COMMAND_OUTPUT+)>

<!ELEMENT COMMAND (ID, STATUS, OPERATION, FRIENDLY_NAME, LAUNCH_DATE,
                   LAST_UPDATED_DATE)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT STATUS  (#PCDATA)>
<!ELEMENT OPERATION (DESCRIPTION, STATUS)>
<!ELEMENT FRIENDLY_NAME (#PCDATA)>
<!ELEMENT LAUNCH_DATE (#PCDATA)>
<!ELEMENT LAST_UPDATED_DATE (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>

<!ELEMENT COMMAND_OUTPUT (ID, STDIN?, STDOUT?, STDERR?)>
<!ELEMENT STDIN (#PCDATA)>
<!ELEMENT STDOUT (#PCDATA)>
<!ELEMENT STDERR (#PCDATA)>

<!ELEMENT WARNING (TEXT, DUPLICATE_HOSTS, URL)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>

<!ELEMENT DUPLICATE_HOSTS (DUPLICATE_HOST*)>

<!ELEMENT DUPLICATE_HOST (IP, DNS_HOSTNAME, NETBIOS_HOSTNAME,
                          LAST_SCANDATE, TRACKING)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT DNS_HOSTNAME (#PCDATA)>
<!ELEMENT NETBIOS_HOSTNAME (#PCDATA)>
<!ELEMENT LAST_SCANDATE (#PCDATA)>
<!ELEMENT TRACKING (#PCDATA)>

<!-- EOF -->
```

# VM - Choose a Priority Level For Each Scan

### Launch/Schedule Scan - New input parameter

Now you can tell us which of your vulnerability scans has the highest priority and should be processed first. You'll do this at the time you launch/schedule your scan. By default, no priority is set. You can choose from nine priority levels with the highest priority being 1 - Emergency and the lowest priority being 9 - Low.

Use the new parameter "priority" when launching/scheduling a scan.

| Parameter | Description |
|---|---|
| priority={value} | (Optional) Specify a value of 0 - 9 to set a processing priority level for the scan. When not specified, a value of 0 (no priority) is used.<br><br>Valid values are:<br>0 = No Priority (the default)<br>1 = Emergency<br>2 = Ultimate<br>3 = Critical<br>4 = Major<br>5 = High<br>6 = Standard<br>7 = Medium<br>8 = Minor<br>9 = Low |

We'll process Finished scans before Running scans, in this order:

- Finished scan with priority set
- Finished scan with no priority
- Running scan with priority set
- Running scan with no priority

### Scan/Schedule List - Output includes processing priority

The output for the Scan List API and Schedule Scan List API was updated to include the processing priority setting for each scan. The PROCESSING_PRIORITY element has been added to the Scan List Output DTD and Schedule Scan List Output DTD.

## Launch Vulnerability Scan

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=launch&scan_title=API_SCAN&ip=10.10.10.28&priority=1&option_title
=Initial+Options"
"https://qualysapi.qualys.com/api/2.0/fo/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>2016-10-04T04:19:21Z</DATETIME>
    <TEXT>New vm scan launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>21123</VALUE>
      </ITEM>
      <ITEM>
        <KEY>REFERENCE</KEY>
        <VALUE>scan/145554761.21123</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

## Schedule Vulnerability Scan

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=create&exclude_ip_per_scan=64.39.96.0-
64.39.111.255&scan_title=API_Scheduled_Scan&priority=2&ip=10.10.10.28&act
ive=1&occurrence=daily&recurrence=1&start_date=04/14/2016&start_hour=17&s
tart_minute=40&end_after=1&time_zone_code=IN&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>2016-10-04T04:16:27Z</DATETIME>
    <TEXT>New scan scheduled successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>24373</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

## Scan List

API request:

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

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCAN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/scan_list_output.dtd">
<SCAN_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2016-10-05T07:06:31Z</DATETIME>
        <SCAN_LIST>
            <SCAN>
                <REF>scan/147565726.06524</REF>
                <TYPE>On-Demand</TYPE>
                <TITLE>
                    <![CDATA[Single IP Scan - 20161005]]>
                </TITLE>
                <USER_LOGIN>qualys_tb</USER_LOGIN>
                <LAUNCH_DATETIME>2016-10-05T05:35:26Z</LAUNCH_DATETIME>
                <DURATION>00:00:06</DURATION>
                <PROCESSING_PRIORITY>2 - Ultimate</PROCESSING_PRIORITY>
                <PROCESSED>1</PROCESSED>
                <STATUS>
                    <STATE>Finished</STATE>
                    <SUB_STATE>No_Host</SUB_STATE>
                </STATUS>
                <TARGET>
                    <![CDATA[10.113.195.148]]>
                </TARGET>
            </SCAN>
        </SCAN_LIST>
    </RESPONSE>
</SCAN_LIST_OUTPUT>
```

DTD update:

The PROCESSING_PRIORITY element has been added to the scan list output DTD (scan_list_output.dtd).

```
<!-- QUALYS SCAN_LIST_OUTPUT DTD -->

<!ELEMENT SCAN_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, SCAN_LIST?)>
<!ELEMENT SCAN_LIST (SCAN+)>
<!ELEMENT SCAN (ID?, REF, TYPE, TITLE, USER_LOGIN, LAUNCH_DATETIME,
                DURATION, PROCESSING_PRIORITY?, PROCESSED, STATUS?,
                TARGET, ASSET_GROUP_TITLE_LIST?, OPTION_PROFILE?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT REF (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT LAUNCH_DATETIME (#PCDATA)>
<!ELEMENT DURATION (#PCDATA)>
<!ELEMENT PROCESSING_PRIORITY (#PCDATA)>
<!ELEMENT PROCESSED (#PCDATA)>
...
```

## Schedule Scan List

API request:

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

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>2016-10-05T07:06:17Z</DATETIME>
    <SCHEDULE_SCAN_LIST>
        <SCAN>
            <ID>9516524</ID>
            <ACTIVE>1</ACTIVE>
            <TITLE>
                <![CDATA[Schedule Scan]]>
            </TITLE>
            <USER_LOGIN>qualys_tb</USER_LOGIN>
            <TARGET>
               <![CDATA[10.10.10.138, 10.113.195.105, 10.113.195.148,
10.113.197.63]]>
            </TARGET>
            <NETWORK_ID>
                <![CDATA[0]]>
            </NETWORK_ID>
            <ISCANNER_NAME>
                <![CDATA[External Scanner]]>
            </ISCANNER_NAME>
            <ASSET_GROUP_TITLE_LIST>
                <ASSET_GROUP_TITLE>
                    <![CDATA[Large]]>
                </ASSET_GROUP_TITLE>
            </ASSET_GROUP_TITLE_LIST>
            <OPTION_PROFILE>
                <TITLE>
                  <![CDATA[Initial Options - Authentication enabled]]>
                </TITLE>
                <DEFAULT_FLAG>1</DEFAULT_FLAG>
            </OPTION_PROFILE>
            <PROCESSING_PRIORITY>3 - Critical</PROCESSING_PRIORITY>
            <SCHEDULE>
                <DAILY frequency_days="1" />
                <START_DATE_UTC>2016-10-05T08:00:00Z</START_DATE_UTC>
                <START_HOUR>0</START_HOUR>
                <START_MINUTE>0</START_MINUTE>
                <PAUSE_AFTER_HOURS></PAUSE_AFTER_HOURS>
                <RESUME_IN_DAYS></RESUME_IN_DAYS>
                <NEXTLAUNCH_UTC>2016-10-05T08:00:00</NEXTLAUNCH_UTC>
                <TIME_ZONE>
                    <TIME_ZONE_CODE>CA-BC</TIME_ZONE_CODE>
                    <TIME_ZONE_DETAILS>(GMT-0800) Canada:
America/Vancouver</TIME_ZONE_DETAILS>
                </TIME_ZONE>
                <DST_SELECTED>0</DST_SELECTED>
            </SCHEDULE>
        </SCAN>
    </SCHEDULE_SCAN_LIST>
```

```
    </RESPONSE>
</SCHEDULE_SCAN_LIST_OUTPUT>
```

DTD update:

The PROCESSING_PRIORITY element has been added to the schedule scan list output DTD (schedule_scan_list_output.dtd).

```
<!-- QUALYS SCHEDULE_SCAN_LIST_OUTPUT DTD -->

<!ELEMENT SCHEDULE_SCAN_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, SCHEDULE_SCAN_LIST?)>
<!ELEMENT SCHEDULE_SCAN_LIST (SCAN+)>
<!ELEMENT SCAN (ID, ACTIVE, TITLE?, USER_LOGIN, TARGET, NETWORK_ID?,
                ISCANNER_NAME?, EC2_INSTANCE?, ASSET_GROUP_TITLE_LIST?,
                ASSET_TAGS?, EXCLUDE_IP_PER_SCAN?, USER_ENTERED_IPS?,
                OPTION_PROFILE?, PROCESSING_PRIORITY?, SCHEDULE)>
...
<!ELEMENT OPTION_PROFILE (TITLE, DEFAULT_FLAG?)>
<!ELEMENT PROCESSING_PRIORITY (#PCDATA)>
<!ELEMENT DEFAULT_FLAG (#PCDATA)>
...
```

# VM - Improvements to Reporting Host Scan Time

### Host scan time is now based on scan end date

We've changed the way we report the host scan time when updating vulnerabilities and tickets. The host scan time will now be based on when the scan finished, not when the scan started. We'll get this date from QID 45038 "Host Scan Time". If this QID was not included in your vulnerability scan then we'll use the scan start date/time.

### New filter parameters

You can now filter the Host List API and Host List VM Detection API based on scan end dates and scan processed dates. Specify the date in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like "2016-09-12" or "2016-09-12T23:15:00Z".

| Parameter | Description |
|---|---|
| vm_processed_before={date} | (Optional) Show hosts with vulnerability scan results processed before a certain date and time (optional). |
| vm_processed_after={date} | (Optional) Show hosts with vulnerability scan results processed after a certain date and time (optional). |
| vm_scan_date_before={date} | (Optional) Show hosts with a vulnerability scan end date before a certain date and time (optional). |
| vm_scan_date_after={date} | (Optional) Show hosts with a vulnerability scan end date after a certain date and time (optional). |
| vm_auth_scan_date_before={date} | (Optional) Show hosts with a successful authenticated vulnerability scan end date before a certain date and time (optional). |
| vm_auth_scan_date_after={date} | (Optional) Show hosts with a successful authenticated vulnerability scan end date after a certain date and time (optional). |

### Output includes scan end dates and duration

The output for the Host List API and Host List VM Detection API was updated to include the following details.

LAST_VM_SCANNED_DATE is the scan end date/time for the most recent unauthenticated vulnerability scan on the host.

LAST_VM_SCANNED_DURATION is the scan duration (in seconds) for the most recent unauthenticated vulnerability scan on the host.

LAST_VM_AUTH_SCANNED_DATE is the scan end date/time for the last successful authenticated vulnerability scan on the host.

LAST_VM_AUTH_SCANNED_DURATION is the scan duration (in seconds) for the last successful authenticated vulnerability scan on the host.

LAST_PC_SCANNED_DATE is the scan end date/time for the most recent compliance scan on the host. (Note - this was only added to the Host List VM Detection API output.)

## Host List API

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/?action=list&truncati
on_limit=10&details=All/AGs&vm_scan_date_before=2016-09-
14T06:32:15Z&vm_auth_scan_date_before=2016-09-
14T06:32:15Z&vm_scan_date_after=2016-05-
12T06:32:15Z&vm_auth_scan_date_after=2016-05-
12T06:32:15Z&vm_processed_before=2016-09-
14T06:33:24Z&vm_processed_after=2016-09-12T06:33:24Z"
```

XML output:

```
...
<HOST_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2016-09-13T09:30:48Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>597</ID>
        <IP>10.10.10.28</IP>
        <TRACKING_METHOD>IP</TRACKING_METHOD>
        <DNS><![CDATA[xpsp3-10-28]]></DNS>
        <NETBIOS><![CDATA[XPSP3-10-28]]></NETBIOS>
        <OS><![CDATA[Windows XP]]></OS>
        <QG_HOSTID><![CDATA[]]></QG_HOSTID>
        <LAST_VULN_SCAN_DATETIME>2016-09-
12T20:16:24Z</LAST_VULN_SCAN_DATETIME>
        <LAST_VM_SCANNED_DATE>2016-09-12T20:16:24Z</LAST_VM_SCANNED_DATE>
        <LAST_VM_SCANNED_DURATION>228</LAST_VM_SCANNED_DURATION>
        <LAST_VM_AUTH_SCANNED_DATE>2016-09-
11T08:18:19Z</LAST_VM_AUTH_SCANNED_DATE>

<LAST_VM_AUTH_SCANNED_DURATION>1150</LAST_VM_AUTH_SCANNED_DURATION>
        <LAST_COMPLIANCE_SCAN_DATETIME>2016-08-
31T02:43:07Z</LAST_COMPLIANCE_SCAN_DATETIME>
        <ASSET_GROUP_IDS>1083,1084</ASSET_GROUP_IDS>
      </HOST>
    </HOST_LIST>
...
```

DTD update:

We added new elements (in bold) to the Host List Output DTD (host_list_output.dtd).

```
...
<!ELEMENT HOST_LIST (HOST+)>
<!ELEMENT HOST (ID, IP?, TRACKING_METHOD?, NETWORK_ID?,
                DNS?, EC2_INSTANCE_ID?, NETBIOS?, OS?, QG_HOSTID?, TAGS?,
                LAST_VULN_SCAN_DATETIME?, LAST_VM_SCANNED_DATE?,
                LAST_VM_SCANNED_DURATION?, LAST_VM_AUTH_SCANNED_DATE?,
                LAST_VM_AUTH_SCANNED_DURATION?,
                LAST_COMPLIANCE_SCAN_DATETIME?, OWNER?, COMMENTS?,
                USER_DEF?, ASSET_GROUP_IDS?)>
...
<!ELEMENT LAST_VULN_SCAN_DATETIME (#PCDATA)>
<!ELEMENT LAST_VM_SCANNED_DATE (#PCDATA)>
<!ELEMENT LAST_VM_SCANNED_DURATION (#PCDATA)>
<!ELEMENT LAST_VM_AUTH_SCANNED_DATE (#PCDATA)>
<!ELEMENT LAST_VM_AUTH_SCANNED_DURATION (#PCDATA)>
<!ELEMENT LAST_COMPLIANCE_SCAN_DATETIME (#PCDATA)>
...
```

## Vulnerability Detection API

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=
list&truncation_limit=10&vm_scan_date_before=2016-09-
14T06:32:15Z&vm_auth_scan_date_before=2016-09-
14T06:32:15Z&vm_scan_date_after=2016-05-
12T06:32:15Z&vm_auth_scan_date_after=2016-05-
12T06:32:15Z&vm_processed_before=2016-09-
14T06:33:24Z&vm_processed_after=2016-09-12T06:33:24Z"
```

XML output:

```
...
<HOST_LIST_VM_DETECTION_OUTPUT>
  <RESPONSE>
    <DATETIME>2016-09-13T09:34:31Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>597</ID>
        <IP>10.10.10.28</IP>
        <TRACKING_METHOD>IP</TRACKING_METHOD>
        <OS><![CDATA[Windows XP]]></OS>
        <OS_CPE><![CDATA[cpe:/o:microsoft:windows_xp::sp3::]]></OS_CPE>
        <DNS><![CDATA[xpsp3-10-28test]]></DNS>
        <NETBIOS><![CDATA[XPSP3-10-28TEST]]></NETBIOS>
```

```
        <LAST_SCAN_DATETIME>2016-09-12T20:20:20Z</LAST_SCAN_DATETIME>
        <LAST_VM_SCANNED_DATE>2016-09-12T20:16:24Z</LAST_VM_SCANNED_DATE>
        <LAST_VM_SCANNED_DURATION>228</LAST_VM_SCANNED_DURATION>
        <LAST_VM_AUTH_SCANNED_DATE>2016-09-
11T08:18:19Z</LAST_VM_AUTH_SCANNED_DATE>

<LAST_VM_AUTH_SCANNED_DURATION>1150</LAST_VM_AUTH_SCANNED_DURATION>
        <LAST_PC_SCANNED_DATE>2016-08-31T02:43:07Z</LAST_PC_SCANNED_DATE>
...
```

DTD update:

We added new elements (in bold) to the Host List VM Detection Output DTD (host_list_vm_detection_output.dtd).

```
...
<!ELEMENT HOST_LIST (HOST+)>
<!ELEMENT HOST (ID, IP?, IPV6?, TRACKING_METHOD?, NETWORK_ID?,
                OS?, OS_CPE?, DNS?, NETBIOS?, QG_HOSTID?,
                LAST_SCAN_DATETIME?, LAST_VM_SCANNED_DATE?,
                LAST_VM_SCANNED_DURATION?, LAST_VM_AUTH_SCANNED_DATE?,
                LAST_VM_AUTH_SCANNED_DURATION?, LAST_PC_SCANNED_DATE?,
                TAGS?, DETECTION_LIST)>
...
<!ELEMENT LAST_SCAN_DATETIME (#PCDATA)>
<!ELEMENT LAST_VM_SCANNED_DATE (#PCDATA)>
<!ELEMENT LAST_VM_SCANNED_DURATION (#PCDATA)>
<!ELEMENT LAST_VM_AUTH_SCANNED_DATE (#PCDATA)>
<!ELEMENT LAST_VM_AUTH_SCANNED_DURATION (#PCDATA)>
<!ELEMENT LAST_PC_SCANNED_DATE (#PCDATA)>
...
```

# VM - More Detection Info Returned from Vulnerability Detection API

The output for the Host List VM Detection API (/api/2.0/fo/asset/host/vm/detection) includes more detection information:

IS_DISABLED is a flag indicating whether the vulnerability is globally disabled for all hosts. A value of 1 means it is disabled, a value of 0 means it is not disabled.

IS_IGNORED is a flag indicating whether the vulnerability is ignored for the particular host. A value of 1 means it is ignored, a value of 0 means it is not ignored.

TIMES_FOUND is the number of times the vulnerability was detected on the host.

SERVICE is the service the vulnerability was detected on, if applicable.

## Vulnerability Detection API

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=
list&truncation_limit=10"
```

XML output:

```
...
        <DETECTION_LIST>
          <DETECTION>
            <QID>38094</QID>
            <TYPE>Potential</TYPE>
            <SEVERITY>2</SEVERITY>
            <PORT>3389</PORT>
            <PROTOCOL>tcp</PROTOCOL>
            <SSL>0</SSL>
            <RESULTS><![CDATA[Detected service win_remote_desktop and os
WINDOWS XP SERVICE PACK 2-3]]></RESULTS>
            <STATUS>Active</STATUS>
            <FIRST_FOUND_DATETIME>2016-08-
01T06:16:09Z</FIRST_FOUND_DATETIME>
            <LAST_FOUND_DATETIME>2016-09-
12T20:16:24Z</LAST_FOUND_DATETIME>
            <LAST_TEST_DATETIME>2016-09-12T20:16:24Z</LAST_TEST_DATETIME>
            <LAST_UPDATE_DATETIME>2016-09-
12T20:20:20Z</LAST_UPDATE_DATETIME>
            <IS_IGNORED>0</IS_IGNORED>
            <IS_DISABLED>0</IS_DISABLED>
            <TIMES_FOUND>74</TIMES_FOUND>
```

```
          </DETECTION>
</DETECTION_LIST>
      </HOST>
    </HOST_LIST>
  </RESPONSE>
</HOST_LIST_VM_DETECTION_OUTPUT>
```

DTD update:

We added new detection elements (in bold) to the Host List VM Detection Output DTD (host_list_vm_detection_output.dtd).

```
...
<!ELEMENT DETECTION_LIST (DETECTION+)>
<!ELEMENT DETECTION (QID, TYPE, SEVERITY?, PORT?, PROTOCOL?, FQDN?, SSL?,
                     INSTANCE?, RESULTS?, STATUS?, FIRST_FOUND_DATETIME?,
                     LAST_FOUND_DATETIME?, LAST_TEST_DATETIME?,
                     LAST_UPDATE_DATETIME?, LAST_FIXED_DATETIME?,
                     IS_IGNORED?, IS_DISABLED?, TIMES_FOUND?, SERVICE?)>
...
<!ELEMENT SERVICE (#PCDATA)>
<!ELEMENT IS_IGNORED (#PCDATA)>
<!ELEMENT IS_DISABLED (#PCDATA)>
<!ELEMENT TIMES_FOUND (#PCDATA)>
...
```

# VM - Easily Identify Disabled Vulnerabilities in KnowledgeBase APIs

We've added a new flag to the XML output of KnowledgeBase APIs to identify vulnerabilities that have been disabled. Managers can disable vulnerabilities in the KnowledgeBase in order to globally filter them from all hosts.

When IS_DISABLED has a value of 1 in the XML output then it is disabled. A value of 0 means it is not disabled.

## KnowledgeBase API (v2)

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

| Parameter | Description |
|---|---|
| show_disabled_flag={**0**\|1} | (Optional) Specify 1 to include the disabled flag for each vulnerability in the XML output. |

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=list&details=Basic&echo_request=0&ids=32&show_disabled_flag=1"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/"
```

XML output:

```
...
    <VULN_LIST>
      <VULN>
        <QID>32</QID>
        <VULN_TYPE>Information Gathered</VULN_TYPE>
        <SEVERITY_LEVEL>2</SEVERITY_LEVEL>
        <TITLE><![CDATA[Darxite Banner]]></TITLE>
        <CATEGORY>General remote services</CATEGORY>
        <LAST_SERVICE_MODIFICATION_DATETIME>1999-01-
01T08:00:00Z</LAST_SERVICE_MODIFICATION_DATETIME>
        <PUBLISHED_DATETIME>2000-11-22T18:29:32Z</PUBLISHED_DATETIME>
        <PATCHABLE>0</PATCHABLE>
        <PCI_FLAG>0</PCI_FLAG>
        <DISCOVERY>
          <REMOTE>1</REMOTE>
        </DISCOVERY>
        <IS_DISABLED>0</IS_DISABLED>
      </VULN>
...
```

DTD update:

We added the IS_DISABLED element to the KnowledgeBase Output DTD (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, PCI_REASONS?, SUPPORTED_MODULES?,
                      DISCOVERY, IS_DISABLED? )>
...
  <!ELEMENT IS_DISABLED (#PCDATA)>
...
```

## KnowledgeBase Download (v1)

Use the new parameter “show_disabled_flag” when downloading the KnowledgeBase to see the disabled flag for each vulnerability.

| Parameter | Description |
|---|---|
| show_disabled_flag={**0** \| 1} | (Optional) Specify 1 to include the disabled flag for each vulnerability in the XML output. |

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"show_disabled_flag=1&vuln_id=118873"
"https://qualysapi.qualys.com/msp/knowledgebase_download.php?"
```

XML output:

```
<VULNS>
  <VULN>
    <QID>118873</QID>
    <VULN_TYPE>Vulnerability</VULN_TYPE>
    <SEVERITY_LEVEL>4</SEVERITY_LEVEL>
    <TITLE><![CDATA[Red Hat Update for apr-util (RHSA-
2010:0950)]]></TITLE>
    ...
    <DISCOVERY>
```

```
    <REMOTE>0</REMOTE>
    <AUTH_TYPE_LIST>
    <AUTH_TYPE>Unix</AUTH_TYPE>
    </AUTH_TYPE_LIST>
      <ADDITIONAL_INFO>Patch Available</ADDITIONAL_INFO>
    </DISCOVERY>
    <IS_DISABLED>0</IS_DISABLED>
  </VULN>
</VULNS>
```

DTD update:

We added the IS_DISABLED element to the KnowledgeBase Download DTD (knowledgebase_download.dtd).

```
...
<!ELEMENT VULN (QID, VULN_TYPE, SEVERITY_LEVEL, TITLE, CATEGORY?,
                DETECTION_INFO?, LAST_UPDATE?,
                BUGTRAQ_ID_LIST?, PATCHABLE, VENDOR_REFERENCE_LIST?,
                CVE_ID_LIST?, DIAGNOSIS?, CONSEQUENCE?, SOLUTION?,
                COMPLIANCE?, CORRELATION?, CVSS_BASE?, CVSS_TEMPORAL?,
                CVSS3_BASE?, CVSS3_TEMPORAL?, CVSS_ACCESS_VECTOR?,
                CVSS_ACCESS_COMPLEXITY?, CVSS_AUTHENTICATION?,
                CVSS_CONFIDENTIALITY_IMPACT?, CVSS_INTEGRITY_IMPACT?,
                CVSS_AVAILABILITY_IMPACT?, CVSS_EXPLOITABILITY?,
                CVSS_REMEDIATION_LEVEL?, CVSS_REPORT_CONFIDENCE?,
                PCI_FLAG?, PCI_REASONS?, SUPPORTED_MODULES?,DISCOVERY?,
                IS_DISABLED?)>
...
<!ELEMENT IS_DISABLED (#PCDATA)>
```

# VM - Removed Version element of CVSS v3

We've updated XML output returned from the KnowledgeBase API (v2) (/api/2.0/fo/knowledge_base/vuln/?action=list) to remove the VERSION sub-element for CVSS_V3 as it is not applicable.

## KnowledgeBase API (v2)

This API no longer returns the VERSION sub-element under CVSS_v3.

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl demo 2' -d
"action=list& ids=105095, 87008,38477"
'https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/'
```

XML output:

```
...
  <VULN_LIST>
            <VULN>
                <QID>105095</QID>
                <VULN_TYPE>Vulnerability</VULN_TYPE>
                <SEVERITY_LEVEL>4</SEVERITY_LEVEL>
                <TITLE>
                    <![CDATA[User(s) With Blank Password]]>
                </TITLE>
                <CATEGORY>Security Policy</CATEGORY>
                <LAST_SERVICE_MODIFICATION_DATETIME>2014-09-
22T17:45:19Z</LAST_SERVICE_MODIFICATION_DATETIME>
                <PUBLISHED_DATETIME>2005-02-
23T08:00:00Z</PUBLISHED_DATETIME>
                <PATCHABLE>0</PATCHABLE>
                <DIAGNOSIS>
                  <![CDATA[The users have the blank password in the shadow
file. These users connect to the system without entering a password.]]>
                </DIAGNOSIS>
                <CONSEQUENCE>
                    <![CDATA[An attacker may connect to the system by
knowing just the username.]]>
                </CONSEQUENCE>
                <SOLUTION>
                    <![CDATA[Set the password for all the users.]]>
                </SOLUTION>
                <CVSS>
                    <BASE source="service">7.5</BASE>
                    <TEMPORAL>7.1</TEMPORAL>
                 </CVSS>
```

```
                <CVSS_V3>
                    <BASE>2.1</BASE>
                    <TEMPORAL>6.5</TEMPORAL>
                </CVSS_V3>
                <PCI_FLAG>1</PCI_FLAG>
                <DISCOVERY>
                    <REMOTE>0</REMOTE>
                    <AUTH_TYPE_LIST>
                        <AUTH_TYPE>Unix</AUTH_TYPE>
                    </AUTH_TYPE_LIST>
                </DISCOVERY>
            </VULN>
        </VULN_LIST>
        ...
```

DTD update:

We removed VERSION element for CVSS_V3 from 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, PCI_REASONS?, SUPPORTED_MODULES?,
                      DISCOVERY)>
...
<!ELEMENT CVSS_V3 (BASE, TEMPORAL?, ACCESS?, IMPACT?,
                   AUTHENTICATION?, EXPLOITABILITY?, REMEDIATION_LEVEL?,
                   REPORT_CONFIDENCE?)>
...
```

# VM - CVSS3 Final Score in Scan Reports

We've added the CVSS3 final score in scan reports with host based findings (also known as asset data reports). Both XML and CSV formats were updated.

### Sample XML Report

```
...
<VULN_INFO>
   <QID id="qid_66021">66021</QID>
   <TYPE>Vuln</TYPE>
   <SSL>false</SSL>
   <RESULT>
<![CDATA[@(#)pcnfsd_v2.c 1.6 - rpc.pcnfsd V2.0 (c) 1991 Sun Technology
Enterprises, Inc.]]>
   </RESULT>
   <FIRST_FOUND>2016-07-13T06:54:15Z</FIRST_FOUND>
   <LAST_FOUND>2016-07-13T06:54:15Z</LAST_FOUND>
   <TIMES_FOUND>1</TIMES_FOUND>
   <VULN_STATUS>New</VULN_STATUS>
   <CVSS_FINAL>4</CVSS_FINAL>
   <CVSS3_FINAL>3.2</CVSS3_FINAL>
</VULN_INFO>
...
```

Updated DTD: asset_data_report.dtd

```
...
<!ELEMENT VULN_INFO (QID, TYPE, PORT?, SERVICE?, FQDN?, PROTOCOL?, SSL?,
INSTANCE?, RESULT?, FIRST_FOUND?, LAST_FOUND?, TIMES_FOUND?,
VULN_STATUS?, LAST_FIXED?, CVSS_FINAL?, CVSS3_FINAL?, TICKET_NUMBER?,
TICKET_STATE?)>
<!ELEMENT QID (#PCDATA)>
<!ATTLIST QID id IDREF #REQUIRED>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT PORT (#PCDATA)>
<!ELEMENT SERVICE (#PCDATA)>
<!ELEMENT FQDN (#PCDATA)>
<!ELEMENT PROTOCOL (#PCDATA)>
<!ELEMENT SSL (#PCDATA)>
<!ELEMENT RESULT (#PCDATA)>
<!ATTLIST RESULT format CDATA #IMPLIED>
<!ELEMENT FIRST_FOUND (#PCDATA)>
<!ELEMENT LAST_FOUND (#PCDATA)>
<!ELEMENT TIMES_FOUND (#PCDATA)>
<!-- Note: VULN_STATUS is N/A for IGs -->
<!ELEMENT VULN_STATUS (#PCDATA)>
<!ELEMENT LAST_FIXED (#PCDATA)>
```

```
<!ELEMENT CVSS_FINAL (#PCDATA)>
<!ELEMENT CVSS3_FINAL (#PCDATA)>
<!ELEMENT TICKET_NUMBER (#PCDATA)>
<!ELEMENT TICKET_STATE (#PCDATA)>
<!ELEMENT INSTANCE (#PCDATA)>
...
```

## Sample CSV Report

```
...

"IP","DNS","NetBIOS","Tracking Method","OS","IP
Status","QID","Title","VulnStatus","Type","Severity","Port","Protocol","F
QDN","SSL","First Detected","Last Detected","Times Detected","CVE
ID","Vendor Reference","Bugtraq ID","CVSS","CVSS Base","CVSS
Temporal","CVSS Environment","CVSS3","CVSS3 Base","CVSS3 Temporal",...

"10.10.24.72","2k3x64sp2-24-72","2K3X64SP2-24-72","IP","Windows 2003 R2
Service Pack 2","host scanned, found vuln","38626","OpenSSL oracle padding
vulnerability(CVE-2016-2107)","New","Vuln","4","2381","tcp",,"over
ssl","08/12/2016 16:34:37","08/12/2016 16:34:37","1","CVE-2016-
2107","OpenSSL Security Advisory 20160503","91787","4.3","2.6
(AV:N/AC:H/Au:N/C:P/I:N/A:N)","2 (E:POC/RL:OF/RC:C)","Asset Group: AG 24,
Collateral Damage Potential:  Medium-High, Target Distribution: Medium,
Confidentiality Requirement: High, Integrity Requirement: Medium,
Availability Requirement: High","5.2","5.8","5.2",...
...
```

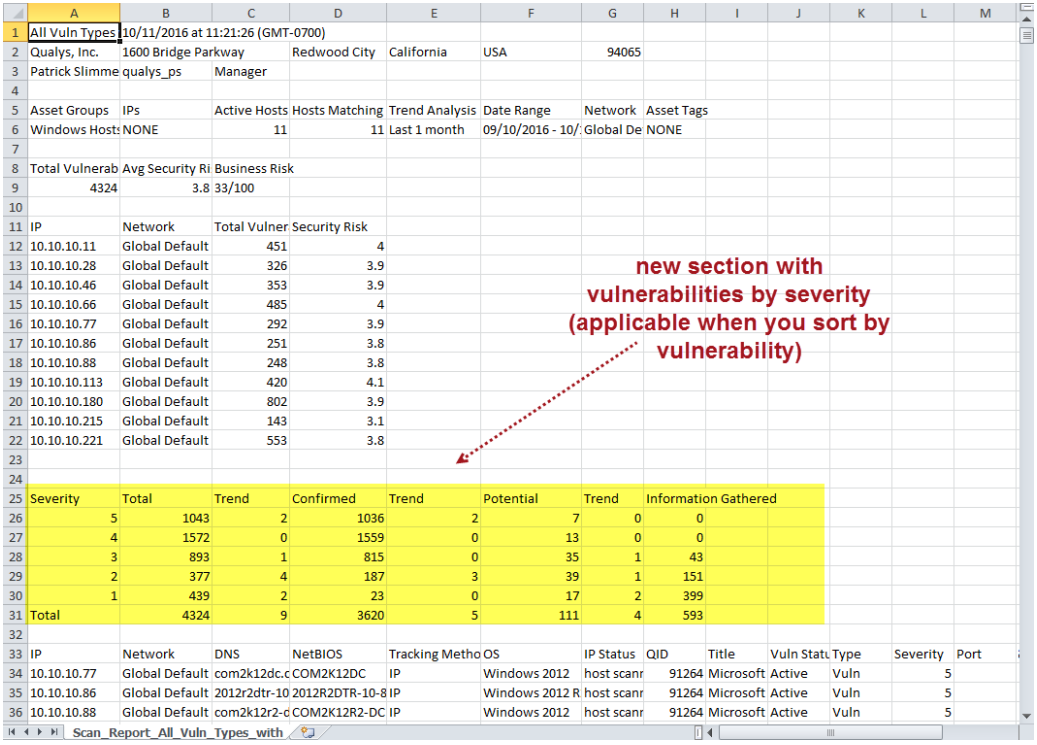
# VM - Vulnerability Counts by Severity Added to Scan Report CSV

This update applies to a scan report with host based findings. Now when you sort your scan report by vulnerability you'll see a section in the CSV output that shows the total number of vulnerabilities detected at each severity level.

For each severity level (1-5), you'll see the total number of vulnerabilities, plus the number of vulnerabilities for each vulnerability type - Confirmed, Potential and Information Gathered (edit your report template to change the types included). When your report includes trending, you'll also see Trend columns showing the changes to vulnerability counts for your report timeframe.

## Sample CSV Report

This sample report includes all vulnerability types - Confirmed, Potential and Information Gathered - and it includes trending for the last month.




| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | All Vuln Types | 10/11/2016 at 11:21:26 (GMT-0700) | | | | | | | | | | | |
| 2 | Qualys, Inc. | 1600 Bridge Parkway | | Redwood City | California | USA | 94065 | | | | | | |
| 3 | Patrick Slimme | qualys_ps | Manager | | | | | | | | | | |
| 4 | | | | | | | | | | | | | |
| 5 | Asset Groups | IPs | Active Hosts | Hosts Matching | Trend Analysis | Date Range | Network | Asset Tags | | | | | |
| 6 | Windows Hosts | NONE | 11 | 11 | Last 1 month | 09/10/2016 - 10/ | Global De | NONE | | | | | |
| 7 | | | | | | | | | | | | | |
| 8 | Total Vulnerab | Avg Security Ri | Business Risk | | | | | | | | | | |
| 9 | 4324 | 3.8 | 33/100 | | | | | | | | | | |
| 10 | | | | | | | | | | | | | |
| 11 | IP | Network | Total Vulner | Security Risk | | | | | | | | | |
| 12 | 10.10.10.11 | Global Default | 451 | 4 | | | | | | | | | |
| 13 | 10.10.10.28 | Global Default | 326 | 3.9 | | | | | | | | | |
| 14 | 10.10.10.46 | Global Default | 353 | 3.9 | | | | | | | | | |
| 15 | 10.10.10.66 | Global Default | 485 | 4 | | | | | | | | | |
| 16 | 10.10.10.77 | Global Default | 292 | 3.9 | | | | | | | | | |
| 17 | 10.10.10.86 | Global Default | 251 | 3.8 | | | | | | | | | |
| 18 | 10.10.10.88 | Global Default | 248 | 3.8 | | | | | | | | | |
| 19 | 10.10.10.113 | Global Default | 420 | 4.1 | | | | | | | | | |
| 20 | 10.10.10.180 | Global Default | 802 | 3.9 | | | | | | | | | |
| 21 | 10.10.10.215 | Global Default | 143 | 3.1 | | | | | | | | | |
| 22 | 10.10.10.221 | Global Default | 553 | 3.8 | | | | | | | | | |
| 23 | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | |
| 25 | Severity | Total | Trend | Confirmed | Trend | Potential | Trend | Information Gathered | | | | | |
| 26 | 5 | 1043 | 2 | 1036 | 2 | 7 | 0 | 0 | | | | | |
| 27 | 4 | 1572 | 0 | 1559 | 0 | 13 | 0 | 0 | | | | | |
| 28 | 3 | 893 | 1 | 815 | 0 | 35 | 1 | 43 | | | | | |
| 29 | 2 | 377 | 4 | 187 | 3 | 39 | 1 | 151 | | | | | |
| 30 | 1 | 439 | 2 | 23 | 0 | 17 | 2 | 399 | | | | | |
| 31 | Total | 4324 | 9 | 3620 | 5 | 111 | 4 | 593 | | | | | |
| 32 | | | | | | | | | | | | | |
| 33 | IP | Network | DNS | NetBIOS | Tracking Metho | OS | IP Status | QID | Title | Vuln Statu | Type | Severity | Port |
| 34 | 10.10.10.77 | Global Default | com2k12dc.c | COM2K12DC | IP | Windows 2012 | host scanr | 91264 | Microsoft | Active | Vuln | 5 | |
| 35 | 10.10.10.86 | Global Default | 2012r2dtr-10 | 2012R2DTR-10-8 | IP | Windows 2012 R | host scanr | 91264 | Microsoft | Active | Vuln | 5 | |
| 36 | 10.10.10.88 | Global Default | com2k12r2-d | COM2K12R2-DC | IP | Windows 2012 | host scanr | 91264 | Microsoft | Active | Vuln | 5 | |

Scan_Report_All_Vuln_Types_with

# VM - Display Last Fixed Date in Scan Reports

When you download a scan report (with host based findings) from your account you'll now see the last fixed date/time for each vulnerability in the report. Download scan reports using any of these methods: download from the UI, use the Report API v2 (/api/2.0/fo/report/?action=fetch), or use the Asset Data Report API v1 (/msp/asset_data_report.php). The Asset Data Report DTD (asset_data_report.dtd) was updated.

XML output:

```
...
<VULN_INFO_LIST>
   <VULN_INFO>
        <QID id="qid_82003">82003</QID>
        <TYPE>Vuln</TYPE>
        <SSL>false</SSL>
        <RESULT>
          <![CDATA[Timestamp of host (host byte ordering): 01:26:12 GMT]]>
        </RESULT>
        <FIRST_FOUND>2016-03-16T05:40:53Z</FIRST_FOUND>
        <LAST_FOUND>2016-07-13T06:54:15Z</LAST_FOUND>
        <TIMES_FOUND>38</TIMES_FOUND>
        <VULN_STATUS>Re-Opened</VULN_STATUS>
        <LAST_FIXED>2016-06-13T12:31:10Z</LAST_FIXED>
        <CVSS_FINAL>-</CVSS_FINAL>
        <TICKET_NUMBER>13669</TICKET_NUMBER>
        <TICKET_STATE>OPEN</TICKET_STATE>
   </VULN_INFO>
...
```

DTD update:

We added the LAST_FIXED element to the Asset Data Report DTD (asset_data_report.dtd).

```
...
<!ELEMENT VULN_INFO (QID, TYPE, PORT?, SERVICE?, FQDN?, PROTOCOL?, SSL?,
                     INSTANCE?, RESULT?, FIRST_FOUND?, LAST_FOUND?,
                     TIMES_FOUND?, VULN_STATUS?, LAST_FIXED?,
                     CVSS_FINAL?, TICKET_NUMBER?, TICKET_STATE?)>
...
<!ELEMENT LAST_FIXED (#PCDATA)>
...
```

# VM - Updates to Vulnerability Scorecard Report

We've made these updates to the Vulnerability Scorecard Report and the Asset Group Scorecard Report DTD (asset_group_scorecard.dtd).

**New Ignored Vulnerabilities Count** - When you choose the new option "Include Vulnerability Ignore Status" in the report template (in the UI), we'll display the number of ignored vulnerabilities in your report. The IGNORED_COUNT element has been added to the Asset Group Scorecard Report DTD.

**Other DTD Updates** - We've added elements to the DTD for vulnerability counts by type (confirmed and potential), by status (new, active, fixed, reopened) and by age (in days). These counts appeared in the report output in previous releases.

You can download scorecard reports from the UI or by using the Report API v2 (/api/2.0/fo/report/?action=fetch).

## Sample XML Report

```
...
  <RESULTS>
    <ASSET_GROUP_LIST>
      <ASSET_GROUP>
        <TITLE><![CDATA[Windows Hosts]]></TITLE>
        <STATS>
          <HOSTS><![CDATA[13]]></HOSTS>
          <NUM_SEV_5><![CDATA[1203]]></NUM_SEV_5>

<NUM_SEV_5_VULNERABLE_HOSTS><![CDATA[13]]></NUM_SEV_5_VULNERABLE_HOSTS>
          <NUM_SEV_4><![CDATA[1860]]></NUM_SEV_4>

<NUM_SEV_4_VULNERABLE_HOSTS><![CDATA[12]]></NUM_SEV_4_VULNERABLE_HOSTS>
          <NUM_SEV_3><![CDATA[955]]></NUM_SEV_3>

<NUM_SEV_3_VULNERABLE_HOSTS><![CDATA[13]]></NUM_SEV_3_VULNERABLE_HOSTS>
          <VULNERABLE_HOSTS><![CDATA[13]]></VULNERABLE_HOSTS>
          <VULNERABLE_HOSTS_PCT><![CDATA[100]]></VULNERABLE_HOSTS_PCT>
          <VULNERABLE_HOSTS_GOAL><![CDATA[0]]></VULNERABLE_HOSTS_GOAL>
          <CONFIRMED_COUNT><![CDATA[4018]]></CONFIRMED_COUNT>
          <POTENTIAL_COUNT><![CDATA[0]]></POTENTIAL_COUNT>
          <NEW_COUNT><![CDATA[0]]></NEW_COUNT>
          <ACTIVE_COUNT><![CDATA[4016]]></ACTIVE_COUNT>
          <FIXED_COUNT><![CDATA[1]]></FIXED_COUNT>
          <REOPENED_COUNT><![CDATA[2]]></REOPENED_COUNT>
          <IGNORED_COUNT><![CDATA[0]]></IGNORED_COUNT>
          <DAY_0_TO_30_COUNT><![CDATA[210]]></DAY_0_TO_30_COUNT>
          <DAY_31_TO_60_COUNT><![CDATA[3808]]></DAY_31_TO_60_COUNT>
          <DAY_61_TO_90_COUNT><![CDATA[0]]></DAY_61_TO_90_COUNT>
```

```
          <DAY_91_TO_180_COUNT><![CDATA[0]]></DAY_91_TO_180_COUNT>
          <DAY_181_TO_270_COUNT><![CDATA[0]]></DAY_181_TO_270_COUNT>
          <DAY_271_TO_365_COUNT><![CDATA[0]]></DAY_271_TO_365_COUNT>
        </STATS>
      </ASSET_GROUP>
...
```

DTD update:

New elements (in bold) were added to the Asset Group Scorecard Report DTD (asset_group_scorecard.dtd).

```
...
<!ELEMENT STATS (HOSTS, NUM_SEV_5?, NUM_SEV_5_VULNERABLE_HOSTS?,
                 NUM_SEV_4?, NUM_SEV_4_VULNERABLE_HOSTS?, NUM_SEV_3?,
                 NUM_SEV_3_VULNERABLE_HOSTS?, VULNERABLE_HOSTS?,
                 VULNERABLE_HOSTS_PCT?,VULNERABLE_HOSTS_GOAL?,
                 CONFIRMED_COUNT?, POTENTIAL_COUNT?, NEW_COUNT?,
                 ACTIVE_COUNT?, FIXED_COUNT?, REOPENED_COUNT?,
                 IGNORED_COUNT?, DAY_0_TO_30_COUNT?, DAY_31_TO_60_COUNT?,
                 DAY_61_TO_90_COUNT?, DAY_91_TO_180_COUNT?,
                 DAY_181_TO_270_COUNT?, DAY_271_TO_365_COUNT?)>
<!ELEMENT HOSTS (#PCDATA)>
<!ELEMENT NUM_SEV_5 (#PCDATA)>
<!ELEMENT NUM_SEV_5_VULNERABLE_HOSTS (#PCDATA)>
<!ELEMENT NUM_SEV_4 (#PCDATA)>
<!ELEMENT NUM_SEV_4_VULNERABLE_HOSTS (#PCDATA)>
<!ELEMENT NUM_SEV_3 (#PCDATA)>
<!ELEMENT NUM_SEV_3_VULNERABLE_HOSTS (#PCDATA)>
<!ELEMENT VULNERABLE_HOSTS (#PCDATA)>
<!ELEMENT VULNERABLE_HOSTS_PCT (#PCDATA)>
<!ELEMENT VULNERABLE_HOSTS_GOAL (#PCDATA)>
<!ELEMENT CONFIRMED_COUNT (#PCDATA)>
<!ELEMENT POTENTIAL_COUNT (#PCDATA)>
<!ELEMENT NEW_COUNT (#PCDATA)>
<!ELEMENT ACTIVE_COUNT (#PCDATA)>
<!ELEMENT FIXED_COUNT (#PCDATA)>
<!ELEMENT REOPENED_COUNT (#PCDATA)>
<!ELEMENT IGNORED_COUNT (#PCDATA)>
<!ELEMENT DAY_0_TO_30_COUNT (#PCDATA)>
<!ELEMENT DAY_31_TO_60_COUNT (#PCDATA)>
<!ELEMENT DAY_61_TO_90_COUNT (#PCDATA)>
<!ELEMENT DAY_91_TO_180_COUNT (#PCDATA)>
<!ELEMENT DAY_181_TO_270_COUNT (#PCDATA)>
<!ELEMENT DAY_271_TO_365_COUNT (#PCDATA)>
...
```

### Sample CSV Report

New columns (in bold) were added to the Asset Group Scorecard Report CSV report.

```
...

"Asset Group Title","Asset Group Hosts Scanned","Severities Level
5","Severities Level 4","Severities Level 3","Severities Level 5
Vulnerable Hosts","Severities Level 4 Vulnerable Hosts","Severities Level
3 Vulnerable Hosts","Vulnerable Hosts Total","Vulnerable Hosts Goal
%","Vulnerable Hosts Goal","Vulnerability Type Confirmed","Vulnerability
Type Potential","Vulnerability Status New","Vulnerability Status
Active","Vulnerability Status Fixed","Vulnerability Status Re-
Opened","Vulnerability Status Ignored","Vulnerability Age (Days)  0-
30","Vulnerability Age (Days)  31-60","Vulnerability Age (Days)  61-
90","Vulnerability Age (Days)  91-180","Vulnerability Age (Days)  181-
270","Vulnerability Age (Days)  > 270"

"Windows
Hosts","13","1203","1860","955","13","12","13","13","100","0","4018","0",
"0","4016","1","2","0","210","3808","0","0","0","0"

...
```

# VM - Scan API v1 Does Not Support Scanning Custom Networks

Using the Scan API v1 (/msp/scan.php) you will now get an error if you try to scan a custom network (i.e. asset groups belonging to a custom network). It's still possible to scan the Global Default Network.

Want to scan custom networks? No problem we've got you covered. Use the Scan API v2 (/api/2.0/fo/scan/). For details, see the Qualys API v2 User Guide.

# VM - Removed PROTOCOL from VULN_INFO for QIDs 38175 and 38228

In scan report XML output, we will no longer show <PROTOCOL>0</PROTOCOL> in the vulnerability details for QID 38175 (reported when an unauthorized service is detected) and QID 38228 (reported when a required service is NOT detected). These QIDs appear in scan reports when you've flagged services as either required or unauthorized in the scan report template.

You can download a scan report (with host based findings) using any of these methods: download from the UI, use the Report API v2 (/api/2.0/fo/report/?action=fetch), or use the Asset Data Report API v1 (/msp/asset_data_report.php). There are no DTD changes.

XML output:

You'll notice that PROTOCOL no longer appears in the output for QIDs 38175 and 38228.

```
...
      <VULN_INFO_LIST>
        <VULN_INFO>
          <QID id="qid_38175">38175</QID>
          <TYPE>Vuln</TYPE>
          <SSL>false</SSL>
          <RESULT><![CDATA[Service &apos;telnet&apos; was detected on
ports 23]]></RESULT>
          <FIRST_FOUND>2016-10-17T18:33:18Z</FIRST_FOUND>
          <LAST_FOUND>2016-10-17T18:33:18Z</LAST_FOUND>
          <TIMES_FOUND>1</TIMES_FOUND>
          <VULN_STATUS>New</VULN_STATUS>
          <CVSS3_FINAL>0</CVSS3_FINAL>
        </VULN_INFO>
        <VULN_INFO>
          <QID id="qid_38228">38228</QID>
          <TYPE>Vuln</TYPE>
          <SSL>false</SSL>
          <RESULT><![CDATA[Service &apos;ActiveSync&apos; was not
detected]]></RESULT>
          <FIRST_FOUND>2016-10-17T18:33:18Z</FIRST_FOUND>
          <LAST_FOUND>2016-10-17T18:33:18Z</LAST_FOUND>
          <TIMES_FOUND>1</TIMES_FOUND>
          <VULN_STATUS>New</VULN_STATUS>
          <CVSS3_FINAL>0</CVSS3_FINAL>
        </VULN_INFO>
...
```

# VM - Created Date Added to Remediation Reports in CSV Format

We've added one new column CreatedDate to Remediation reports in CSV format. The CreatedDate column shows the date/time when the report was created in the new CreatedDate column.

### Sample CSV Report

This sample remediation report was created on October 24, 2016.

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Company | User | ReportTitle | AssetGroups | IPs | Users | CreatedDate | Network | AssetTags | | | |
| 2 | Qualys, Inc. | Patrick Slimmer | Tickets per Vuln | All | | All Users | 10/24/2016 at 12:02:06 (GMT-0700) | Global Default Network | | | | |
| 3 | QID | Title | Type | Disabled | Severity | OriginalSe | Tickets | Open | Resolved | Closed | AvgResolu | Overdue |
| 4 | 90783 | Microsoft Windo | Confirmed | no | 5 | | 5 | 5 | 0 | 0 | N/A | 1 |
| 5 | 90464 | Microsoft Windo | Confirmed | no | 5 | | 4 | 2 | 0 | 2 | 0.5 | 1 |
| 6 | 90477 | Microsoft SMB Re | Confirmed | no | 5 | | 4 | 4 | 0 | 0 | N/A | 1 |
| 7 | 90517 | Microsoft Windo | Confirmed | no | 5 | | 4 | 4 | 0 | 0 | N/A | 0 |
| 8 | 90572 | Microsoft WordP | Confirmed | no | 5 | | 4 | 4 | 0 | 0 | N/A | 0 |
| 9 | 90577 | Microsoft SMB Cl | Confirmed | no | 5 | | 4 | 4 | 0 | 0 | N/A | 0 |
| 10 | 90596 | Microsoft Windo | Confirmed | no | 5 | | 4 | 4 | 0 | 0 | N/A | 0 |
| 11 | 90606 | Microsoft Media | Confirmed | no | 5 | | 4 | 4 | 0 | 0 | N/A | 0 |
| 12 | 90616 | Microsoft Windo | Confirmed | no | 5 | | 4 | 4 | 0 | 0 | N/A | 0 |
| 13 | 90923 | Microsoft Cumul | Confirmed | no | 5 | | 4 | 4 | 0 | 0 | N/A | 0 |

Date/time when the report was created

# PC - Support Asset Tags in Compliance Policies

With this release users have the ability to assign asset tags to their compliance policies using the Policy Editor (in the Qualys UI). Hosts that match any of the included tags will be included in the policy. We've updated the Compliance Policy List output to show the tag set for each policy as well as tag information (ID and name) in the Glossary section. We'll also show whether or not the Evaluate Now option was selected in the policy. (Managers and Auditors have permission to add asset tags to policies.)

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=list&ids=991742279"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/"
```

XML output:

```
<POLICY_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2016-09-08T23:36:40Z</DATETIME>
    <POLICY_LIST>
      <POLICY>
        <ID>11431</ID>
        <TITLE><![CDATA[My WinXP Hosts]]></TITLE>
    ...
        <EVALUATE_NOW><![CDATA[no]]></EVALUATE_NOW>
        <ASSET_GROUP_IDS>1280</ASSET_GROUP_IDS>
        <TAG_SET_INCLUDE>
          <TAG_ID>7508422</TAG_ID>
        </TAG_SET_INCLUDE>
        <TAG_INCLUDE_SELECTOR>ANY</TAG_INCLUDE_SELECTOR>
...
    <GLOSSARY>
      <ASSET_GROUP_LIST>
        <ASSET_GROUP>
          <ID>1280</ID>
          <TITLE><![CDATA[Windows]]></TITLE>
        </ASSET_GROUP>
      </ASSET_GROUP_LIST>
      <ASSET_TAG_LIST>
        <TAG>
          <TAG_ID>7508422</TAG_ID>
          <TAG_NAME>Cloud Agent</TAG_NAME>
        </TAG>
      </ASSET_TAG_LIST>
    </GLOSSARY>
  </RESPONSE>
</POLICY_LIST_OUTPUT>
```

DTD update:

We added the EVALUATE_NOW element plus tag specific elements (in bold) to the Policy List Output DTD (policy_list_output.dtd).

```
...
<!ELEMENT POLICY (ID, TITLE, CREATED?, LAST_MODIFIED?, LAST_EVALUATED?,
                  STATUS?, EVALUATE_NOW?, ASSET_GROUP_IDS?,
                  TAG_SET_INCLUDE?, TAG_INCLUDE_SELECTOR?,
                  INCLUDE_AGENT_IPS?, CONTROL_LIST?)>
...
<!ELEMENT EVALUATE_NOW (#PCDATA)>
...
<!ELEMENT TAG_SET_INCLUDE (TAG_ID+)>
<!ELEMENT TAG_ID (#PCDATA)>
<!ELEMENT TAG_INCLUDE_SELECTOR (#PCDATA)>
...
<!ELEMENT GLOSSARY (ASSET_GROUP_LIST?, ASSET_TAG_LIST?, USER_LIST?)>
...

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

# PC - Include UDCs in Policy Export/Import

You can now include user-defined controls (UDCs) when you export a policy from your account to CSV or XML, and when you import a policy to your account from XML. By default, only service-defined controls are included during policy export and import.

## Export a policy to XML

You can export a compliance policy, that exists in your account, to an XML file. Specify the new input parameter "show_user_controls=1" to include UDCs.

| Parameter | Description |
|---|---|
| show_user_controls={**0** \| 1} | (Optional) When not specified, user-defined controls are not included in the output. Specify **show_user_controls=1** to include UDCs. |

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=export&ids=991742279&show_user_controls=1"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/"
```

XML output:

```
...
        <SECTION>
            <NUMBER>1</NUMBER>
            <HEADING><![CDATA[UDCs]]></HEADING>
            <CONTROLS total="2">
                <USER_DEFINED_CONTROL>
                    <CHECK_TYPE>Registry Key Existence</CHECK_TYPE>
                    <CATEGORY>
                        <ID>3</ID>
                      <NAME><![CDATA[Access Control Requirements]]></NAME>
                    </CATEGORY>
                    <SUB_CATEGORY>
                        <ID>1007</ID>
                        <NAME><![CDATA[Authentication/Passwords]]></NAME>
                    </SUB_CATEGORY>
                    <STATEMENT><![CDATA[check for registry key
HKLM\SYSTEM]]></STATEMENT>
                    <CRITICALITY>
                        <LABEL><![CDATA[MEDIUM]]></LABEL>
                        <VALUE>2</VALUE>
                    </CRITICALITY>
                    <COMMENT><![CDATA[]]></COMMENT>
                    <IGNORE_ERROR>0</IGNORE_ERROR>
```

```
                    <IGNORE_ITEM_NOT_FOUND>0</IGNORE_ITEM_NOT_FOUND>
                    <SCAN_PARAMETERS>
                        <REG_HIVE><![CDATA[HKEY_CLASSES_ROOT
(HKCR)]]></REG_HIVE>
                        <REG_KEY><![CDATA[SYSTEM]]></REG_KEY>
                        <DATA_TYPE>Boolean</DATA_TYPE>
                        <DESCRIPTION><![CDATA[check for registry key
HKLM\SYSTEM]]></DESCRIPTION>
                    </SCAN_PARAMETERS>
                    <TECHNOLOGIES total="1">
                        <TECHNOLOGY>
                            <ID>1</ID>
                            <NAME>Windows XP desktop</NAME>

<EVALUATE><CTRL><DP><K>custom.reg_key_exist.1004002</K><L>2</L><V>true</V
></DP></CTRL></EVALUATE>
                            <RATIONALE><![CDATA[pass if registry key
HKLM\SYSTEM exists]]></RATIONALE>
                            <DATAPOINT>
                                <CARDINALITY>no cd</CARDINALITY>
                                <OPERATOR>no op</OPERATOR>
                                <DEFAULT_VALUES total="1">
                                    <DEFAULT_VALUE>true</DEFAULT_VALUE>
                                </DEFAULT_VALUES>
                            </DATAPOINT>
                        </TECHNOLOGY>
                    </TECHNOLOGIES>
                    <REFERENCE_LIST/>
                </USER_DEFINED_CONTROL>
...
```

DTD update:

We added UDC specific elements (in bold) to the Policy Export Output DTD (policy_export_output.dtd).

```
...
<!ELEMENT CONTROLS ((CONTROL|USER_DEFINED_CONTROL)*)>
<!ATTLIST CONTROLS total CDATA #IMPLIED>
<!ELEMENT CONTROL (ID, CRITICALITY?, REFERENCE_TEXT?, TECHNOLOGIES)>
<!ELEMENT ID (#PCDATA)>

<!ELEMENT STATUS (#PCDATA)>
<!ELEMENT CRITICALITY (LABEL, VALUE)>
<!ELEMENT REFERENCE_TEXT (#PCDATA)>
<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT TECHNOLOGIES (TECHNOLOGY*)>
<!ATTLIST TECHNOLOGIES total CDATA #IMPLIED>
<!ELEMENT TECHNOLOGY (ID, NAME?, EVALUATE?, RATIONALE?, DATAPOINT?)>
```

```
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT EVALUATE (CTRL*)>
<!ELEMENT RATIONALE (#PCDATA)>
<!ELEMENT CTRL (AND|OR|NOT|DP)+>
<!ELEMENT AND (AND|OR|NOT|DP)+>
<!ELEMENT OR (AND|OR|NOT|DP)+>
<!ELEMENT NOT (AND|OR|NOT|DP)+>
<!ELEMENT DP (K|OP|CD|L|V|FV)+>
<!ELEMENT K (#PCDATA)>
<!ELEMENT OP (#PCDATA)>
<!ELEMENT CD (#PCDATA)>
<!ELEMENT L (#PCDATA)>
<!ELEMENT V (#PCDATA)>
<!ELEMENT FV (#PCDATA)>
<!ATTLIST FV set CDATA #IMPLIED>

<!ELEMENT DATAPOINT (CARDINALITY?, OPERATOR?, DEFAULT_VALUES?)>
<!ELEMENT CARDINALITY (#PCDATA)>
<!ELEMENT OPERATOR (#PCDATA)>
<!ELEMENT DEFAULT_VALUES (DEFAULT_VALUE*)>
<!ATTLIST DEFAULT_VALUES total CDATA #IMPLIED>
<!ELEMENT DEFAULT_VALUE (#PCDATA)>

<!ELEMENT USER_DEFINED_CONTROL (CHECK_TYPE, CATEGORY, SUB_CATEGORY,
                                STATEMENT, CRITICALITY?, COMMENT?,
                                IGNORE_ERROR, IGNORE_ITEM_NOT_FOUND?,
                                SCAN_PARAMETERS, REFERENCE_TEXT?,
                                TECHNOLOGIES, REFERENCE_LIST)>
<!ELEMENT CHECK_TYPE (#PCDATA)>

<!ELEMENT CATEGORY (ID, NAME)>
<!ELEMENT SUB_CATEGORY (ID, NAME)>

<!ELEMENT STATEMENT (#PCDATA)>
<!ELEMENT COMMENT (#PCDATA)>
<!ELEMENT IGNORE_ERROR (#PCDATA)>
<!ELEMENT IGNORE_ITEM_NOT_FOUND (#PCDATA)>
<!ELEMENT REFERENCE_LIST (REFERENCE*)>
<!ELEMENT REFERENCE (REF_DESCRIPTION?, URL?)>
<!ELEMENT REF_DESCRIPTION (#PCDATA)>
<!ELEMENT URL (#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?,
                           BASE_DIR?, SHOULD_DESCEND?, DEPTH_LIMIT?,
                           FOLLOW_SYMLINK?, FILE_NAME_MATCH?,
                           FILE_NAME_SKIP?, DIR_NAME_MATCH?,
                           DIR_NAME_SKIP?, PERMISSIONS?, PERM_COND?,
```

```
                              TYPE_MATCH?, USER_OWNER?, GROUP_OWNER?,
                              TIME_LIMIT?, MATCH_LIMIT?,
                              WIN_FILE_SYS_OBJECT_TYPES?,
                              MATCH_WELL_KNOWN_USERS_FOR_ANY_DOMAIN?,
                              WIN_PERMISSION_USERS?, WIN_PERMISSION_MATCH?,
                              WIN_PERMISSIONS?, GROUP_NAME?,
                              GROUP_NAME_LIMIT?, DATA_TYPE, DESCRIPTION)>
<!ELEMENT REG_HIVE (#PCDATA)>
<!ELEMENT REG_KEY (#PCDATA)>
<!ELEMENT REG_VALUE_NAME (#PCDATA)>
<!ELEMENT FILE_PATH (#PCDATA)>
<!ELEMENT FILE_QUERY (#PCDATA)>
<!ELEMENT HASH_TYPE (#PCDATA)>
<!ELEMENT WMI_NS (#PCDATA)>
<!ELEMENT WMI_QUERY (#PCDATA)>
<!ELEMENT SHARE_USER (#PCDATA)>
<!ELEMENT PATH_USER (#PCDATA)>
<!ELEMENT BASE_DIR (#PCDATA)>
<!ELEMENT SHOULD_DESCEND (#PCDATA)>
<!ELEMENT DEPTH_LIMIT (#PCDATA)>
<!ELEMENT FOLLOW_SYMLINK (#PCDATA)>
<!ELEMENT FILE_NAME_MATCH (#PCDATA)>
<!ELEMENT FILE_NAME_SKIP (#PCDATA)>
<!ELEMENT DIR_NAME_MATCH (#PCDATA)>
<!ELEMENT DIR_NAME_SKIP (#PCDATA)>
<!ELEMENT PERM_COND (#PCDATA)>
<!ELEMENT TYPE_MATCH (#PCDATA)>
<!ELEMENT USER_OWNER (#PCDATA)>
<!ELEMENT GROUP_OWNER (#PCDATA)>
<!ELEMENT TIME_LIMIT (#PCDATA)>
<!ELEMENT MATCH_LIMIT (#PCDATA)>
<!ELEMENT WIN_PERMISSION_MATCH (#PCDATA)>
<!ELEMENT MATCH_WELL_KNOWN_USERS_FOR_ANY_DOMAIN (#PCDATA)>
<!ELEMENT WIN_PERMISSION_USERS (#PCDATA)>
<!ELEMENT GROUP_NAME (#PCDATA)>
<!ELEMENT GROUP_NAME_LIMIT (#PCDATA)>
<!ELEMENT DATA_TYPE (#PCDATA)>

<!ELEMENT PERMISSIONS (SPECIAL, USER, GROUP, OTHER)>
<!ELEMENT SPECIAL (SPECIAL_USER, SPECIAL_GROUP, SPECIAL_DELETION)>
<!ELEMENT SPECIAL_USER (#PCDATA)>
<!ELEMENT SPECIAL_GROUP (#PCDATA)>
<!ELEMENT SPECIAL_DELETION (#PCDATA)>

<!ELEMENT USER (READ, WRITE, EXECUTE)>
<!ELEMENT GROUP (READ, WRITE, EXECUTE)>
<!ELEMENT OTHER (READ, WRITE, EXECUTE)>
<!ELEMENT READ (#PCDATA)>
<!ELEMENT WRITE (#PCDATA)>
```

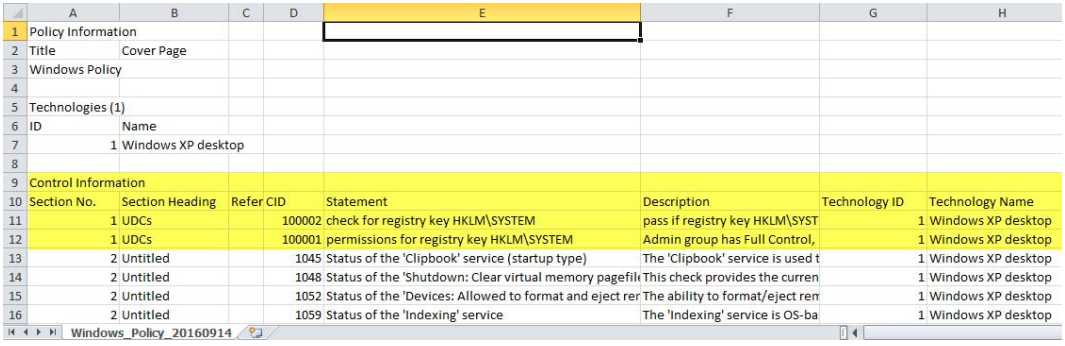
```
<!ELEMENT EXECUTE (#PCDATA)>

<!ELEMENT WIN_PERMISSIONS (WIN_BASIC_PERMISSIONS?,
                           WIN_ADVANCED_PERMISSIONS?)>
<!ELEMENT WIN_BASIC_PERMISSIONS (WIN_BASIC_PERMISSION_TYPE+)>
<!ELEMENT WIN_BASIC_PERMISSION_TYPE (#PCDATA)>
<!ELEMENT WIN_ADVANCED_PERMISSIONS (WIN_ADVANCED_PERMISSION_TYPE+)>
<!ELEMENT WIN_ADVANCED_PERMISSION_TYPE (#PCDATA)>

<!ELEMENT WIN_FILE_SYS_OBJECT_TYPES (#PCDATA)>

<!ELEMENT APPENDIX (OP_ACRONYMS, DATA_POINT_ACRONYMS+)>
<!ELEMENT OP_ACRONYMS (OP+)>
<!ATTLIST OP id CDATA #IMPLIED>
<!ELEMENT DATA_POINT_ACRONYMS (DP+)>
<!ATTLIST K id CDATA #IMPLIED>
<!ATTLIST FV id CDATA #IMPLIED>

<!-- EOF -->
```

## Export a policy to CSV (UI only)

You can export a policy to CSV format from the Qualys UI. Select the option “Include user defined controls” to include UDCs in the output. The user-defined controls are listed under Control Information just like service-defined controls.

***No new columns are added to the CSV report for UDCs if you select the option “Include user defined controls” when exporting a policy from the Qualys UI.

CSV output:



| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Policy Information | | | | | | | |
| 2 | Title | Cover Page | | | | | | |
| 3 | Windows Policy | | | | | | | |
| 4 | | | | | | | | |
| 5 | Technologies (1) | | | | | | | |
| 6 | ID | Name | | | | | | |
| 7 | 1 | Windows XP desktop | | | | | | |
| 8 | | | | | | | | |
| 9 | Control Information | | | | | | | |
| 10 | Section No. | Section Heading | Refer | CID | Statement | Description | Technology ID | Technology Name |
| 11 | 1 | UDCs | | 100002 | check for registry key HKLM\SYSTEM | pass if registry key HKLM\SYST | 1 | Windows XP desktop |
| 12 | 1 | UDCs | | 100001 | permissions for registry key HKLM\SYSTEM | Admin group has Full Control, | 1 | Windows XP desktop |
| 13 | 2 | Untitled | | 1045 | Status of the 'Clipbook' service (startup type) | The 'Clipbook' service is used t | 1 | Windows XP desktop |
| 14 | 2 | Untitled | | 1048 | Status of the 'Shutdown: Clear virtual memory pagefil | This check provides the curren | 1 | Windows XP desktop |
| 15 | 2 | Untitled | | 1052 | Status of the 'Devices: Allowed to format and eject rer | The ability to format/eject ren | 1 | Windows XP desktop |
| 16 | 2 | Untitled | | 1059 | Status of the 'Indexing' service | The 'Indexing' service is OS-ba | 1 | Windows XP desktop |

Windows_Policy_20160914

## Import a policy

You can import a compliance policy, defined in an XML file, into your account. Specify the new input parameter "create_user_controls=1" to include UDCs in the policy.

| Parameter | Description |
|---|---|
| create_user_controls={**0**\|1} | (Optional) When not specified, user-defined controls are not created when you import a policy. Specify **create_user_controls=1** to include UDCs from the XML file. |

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl Sample"
-H "Content-type:text/xml"
--data-binary @policy.xml
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?action=import
&title=My+Policy&create_user_controls=1"
```

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>2016-09-09T21:32:40Z</DATETIME>
      <TEXT>Successfully imported compliance policy</TEXT>
      <ITEM_LIST>
         <ITEM>
           <KEY>ID</KEY>
           <VALUE>136992</VALUE>
         </ITEM>
         <ITEM>
            <KEY>TITLE</KEY>
            <VALUE>My Policy</VALUE>
         </ITEM>
      </ITEM_LIST>
   </RESPONSE>
</SIMPLE_RETURN>
```

# PC - Expose Human Readable Look-ups for Control Descriptions via API

The Compliance Policy Export API (/api/2.0/fo/compliance/policy/?action=export) now includes a new appendix with human readable look-ups for control descriptions, giving you explanation on the various aspects of control description and evaluation, when the request includes show_appendix=1.

| Parameter | Description |
|---|---|
| action=export | (Required) Specifies the action type for exporting the policy. |
| show_appendix = {**0** \| 1} | (Optional) Show the appendix section in the XML output. When not specified, the appendix section is not included in the XML output. Specify 1 to view parameters in the XML output. |

DTD update:

We added a new section for the APPENDIX element to policy_export_output.dtd. You must remove the APPENDIX section if you wish to import this XML as policy.

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

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
POST_DATA?)>
...
<!ELEMENT V (#PCDATA)>
<!ELEMENT FV (#PCDATA)>
<!ATTLIST FV set CDATA #IMPLIED>


<!ELEMENT APPENDIX (OP_ACRONYMS, DATA_POINT_ACRONYMS+)>
<!ELEMENT OP_ACRONYMS (OP+)>
<!ATTLIST OP id CDATA #IMPLIED>
<!ELEMENT DATA_POINT_ACRONYMS (DP+)>
<!ATTLIST K id CDATA #IMPLIED>
<!ATTLIST FV id CDATA #IMPLIED>

<!-- EOF -->
```

API request:

```
curl -u "USERNAME:PASSWORD" GET -H "X-Requested-With: curl" -X "POST" -d
"action=export&id=5438&show_appendix=1"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/">showApp.xml
```

XML output :

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_EXPORT_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_export_
output.dtd">
<POLICY_EXPORT_OUTPUT>
  <RESPONSE>
    <DATETIME>2016-09-09T09:07:13Z</DATETIME>
   <POLICY>
    <TITLE><![CDATA[Solaris]]></TITLE>
    <EXPORTED><![CDATA[2016-09-09T09:07:12Z]]></EXPORTED>
    <COVER_PAGE><![CDATA[]]></COVER_PAGE>
    <STATUS><![CDATA[active]]></STATUS>
    <TECHNOLOGIES total="4">
        <TECHNOLOGY>
            <ID>4</ID>
            <NAME>Solaris 9.x</NAME>
        </TECHNOLOGY>
...
   <SECTION>
            <NUMBER>3</NUMBER>
            <HEADING><![CDATA[Untitled]]></HEADING>
            <CONTROLS total="4"/>
        </SECTION>
    </SECTIONS>
    <!--Note : Remove APPENDIX section if you wish to import this
     XML as policy.-->
    <APPENDIX>
        <OP_ACRONYMS><OP id="lt">less than</OP>
                          <OP id="gt">greater than</OP>
                          <OP id="le">less than or equal to</OP>
                          <OP id="ge">greater than or equal to</OP>
                          <OP id="ne">not equal to</OP>
                         <OP id="xeq">list OR string list</OP>
                          <OP id="eq">equal to</OP>
                          <OP id="in">in</OP>
                          <OP id="xre">regular expression list</OP>
                          <OP id="re">regular expression</OP>
                          <OP id="range">in range</OP></OP_ACRONYMS>
        <DATA_POINT_ACRONYMS>
            <DP>
               <K id="auth.useraccount.legacy-plus-accounts"><![CDATA[The
                following List String value(s) <B>X</B> indicate the
                current list of accounts defined within the <B>/etc/group
                </B>, <B>/etc/shadow</B>, and/or <B>/etc/passwd</B> files
                having a <B>plus-sign '+'</B> preceding them.]]></K>
              <FV id="161803399999999"><![CDATA[Setting not found]]></FV>
              <FV id="314159265358979"><![CDATA[File not found]]></FV>
```

```
                </DP>
                <DP>
                    <K id="auth.useraccount.minimum-password-length">
                      <![CDATA[This Integer value <B>X</B> indicates the
                      current status of the <B>PASSLENGTH 'minimum password
                      length'</B> setting within the <B>/etc/default/passwd
                      </B> file.]]></K>
                   <FV id="161803399999999"><![CDATA[Setting not found]]></FV>
                    <FV id="314159265358979"><![CDATA[File not found]]></FV>
                </DP>
      ...
    </DATA_POINT_ACRONYMS>
    </APPENDIX>
</POLICY>
  </RESPONSE>
</POLICY_EXPORT_OUTPUT>
```

# PC - Policy List Output - added Locked indicator

With this release Managers and Unit Managers have the ability to lock compliance policies. When locked, the policy settings cannot be edited by other users. The output for the Compliance Policy List API (/api/2.0/fo/compliance/policy/ with action=list) has been updated to indicate when a policy is locked.

IS_LOCKED is a flag indicating whether the policy is locked. A value of 1 means it is locked, a value of 0 means it is not locked.

API request:

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

XML output:

```
<POLICY_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2016-09-20T23:17:13Z</DATETIME>
    <POLICY_LIST>
      <POLICY>
        <ID>9429</ID>
        <TITLE><![CDATA[windows]]></TITLE>
        <CREATED>
          <DATETIME>2016-08-23T17:17:47Z</DATETIME>
          <BY>qualys_user</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2016-08-24T20:58:11Z</DATETIME>
          <BY>qualys_user</BY>
        </LAST_MODIFIED>
        <LAST_EVALUATED>
          <DATETIME>2016-09-15T23:38:31Z</DATETIME>
        </LAST_EVALUATED>
        <STATUS><![CDATA[active]]></STATUS>
        <IS_LOCKED>0</IS_LOCKED>
        <EVALUATE_NOW><![CDATA[no]]></EVALUATE_NOW>
        <ASSET_GROUP_IDS>1280</ASSET_GROUP_IDS>
        <CONTROL_LIST>
          <CONTROL>
            <ID>1045</ID>
           <STATEMENT><![CDATA[Status of the 'Clipbook' service (startup
type)]]></STATEMENT>
            <CRITICALITY>
              <LABEL><![CDATA[SERIOUS]]></LABEL>
              <VALUE>3</VALUE>
            </CRITICALITY>
```

```
          </CONTROL>
          <CONTROL>
            <ID>1048</ID>
            <STATEMENT><![CDATA[Status of the 'Shutdown: Clear virtual
memory pagefile' setting]]></STATEMENT>
            <CRITICALITY>
              <LABEL><![CDATA[CRITICAL]]></LABEL>
              <VALUE>4</VALUE>
            </CRITICALITY>
          </CONTROL>
        </CONTROL_LIST>
      </POLICY>
...
```

DTD update:

We added the IS_LOCKED element to the Policy List Output DTD (policy_list_output.dtd).

```
...
<!ELEMENT POLICY (ID, TITLE, CREATED?, LAST_MODIFIED?, LAST_EVALUATED?,
                  STATUS?, IS_LOCKED?, EVALUATE_NOW?, ASSET_GROUP_IDS?,
                  TAG_SET_INCLUDE?, TAG_INCLUDE_SELECTOR?,
                  INCLUDE_AGENT_IPS?, CONTROL_LIST?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>

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

<!ELEMENT LAST_MODIFIED (DATETIME, BY)>

<!ELEMENT LAST_EVALUATED (DATETIME)>

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

# PC - Control List Output - added UDC settings

The control list may include service-defined controls and user-defined controls (UDCs). The XML output has been updated to include settings defined for each UDC, including scan parameter settings, ignore options, datapoint, etc.

DTD update:

Updated the Control List Output DTD (control_list_output.dtd) to include new elements: CHECK_TYPE, COMMENT, IGNORE_ERROR, IGNORE_ITEM_NOT_FOUND, SCAN PARAMETERS (with sub-elements) and DATAPOINT (with sub-elements).

```
<!-- QUALYS CONTROL_LIST_OUTPUT DTD -->
<!ELEMENT CONTROL_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, (CONTROL_LIST|ID_SET)?, WARNING?)>
<!ELEMENT CONTROL_LIST (CONTROL+)>
<!ELEMENT CONTROL (ID, UPDATE_DATE, CREATED_DATE, CATEGORY, SUB_CATEGORY,
STATEMENT, CRITICALITY?, DEPRECATED?, DEPRECATED_DATE?,
CHECK_TYPE?, COMMENT?, IGNORE_ERROR?, IGNORE_ITEM_NOT_FOUND?,
SCAN_PARAMETERS?, TECHNOLOGY_LIST, FRAMEWORK_LIST?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT UPDATE_DATE (#PCDATA)>
<!ELEMENT CREATED_DATE (#PCDATA)>
<!ELEMENT CATEGORY (#PCDATA)>
<!ELEMENT SUB_CATEGORY (#PCDATA)>
<!ELEMENT STATEMENT (#PCDATA)>
<!ELEMENT CRITICALITY (LABEL, VALUE)>
<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT DEPRECATED (#PCDATA)>
<!ELEMENT DEPRECATED_DATE (#PCDATA)>
<!ELEMENT CHECK_TYPE (#PCDATA)>
<!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?, 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?,
DATA_TYPE, DESCRIPTION)>
<!ELEMENT REG_HIVE (#PCDATA)>
<!ELEMENT REG_KEY (#PCDATA)>
<!ELEMENT REG_VALUE_NAME (#PCDATA)>
<!ELEMENT FILE_PATH (#PCDATA)>
<!ELEMENT FILE_QUERY (#PCDATA)>
<!ELEMENT HASH_TYPE (#PCDATA)>
<!ELEMENT WMI_NS (#PCDATA)>
<!ELEMENT WMI_QUERY (#PCDATA)>
<!ELEMENT SHARE_USER (#PCDATA)>
<!ELEMENT PATH_USER (#PCDATA)>
<!ELEMENT GROUP_NAME (#PCDATA)>
<!ELEMENT GROUP_NAME_LIMIT (#PCDATA)>
<!ELEMENT BASE_DIR (#PCDATA)>
<!ELEMENT DEPTH_LIMIT (#PCDATA)>
<!ELEMENT FILE_NAME_MATCH (#PCDATA)>
<!ELEMENT FILE_NAME_SKIP (#PCDATA)>
<!ELEMENT DIR_NAME_MATCH (#PCDATA)>
<!ELEMENT DIR_NAME_SKIP (#PCDATA)>
<!ELEMENT TIME_LIMIT (#PCDATA)>
<!ELEMENT MATCH_LIMIT (#PCDATA)>
<!ELEMENT WIN_FILE_SYS_OBJECT_TYPES (#PCDATA)>
<!ELEMENT MATCH_WELL_KNOWN_USERS_FOR_ANY_DOMAIN (#PCDATA)>
<!ELEMENT WIN_PERMISSION_USERS (#PCDATA)>
<!ELEMENT WIN_PERMISSION_MATCH (#PCDATA)>
<!ELEMENT SHOULD_DESCEND (#PCDATA)>
<!ELEMENT FOLLOW_SYMLINK (#PCDATA)>
<!ELEMENT PERMISSIONS (SPECIAL, USER, GROUP, OTHER)>
<!ELEMENT PERM_COND (#PCDATA)>
<!ELEMENT TYPE_MATCH (#PCDATA)>
<!ELEMENT USER_OWNER (#PCDATA)>
<!ELEMENT GROUP_OWNER (#PCDATA)>

<!ELEMENT WIN_PERMISSIONS (WIN_BASIC_PERMISSIONS?,
WIN_ADVANCED_PERMISSIONS?)>
<!ELEMENT WIN_BASIC_PERMISSIONS (WIN_BASIC_PERMISSION_TYPE+)>
<!ELEMENT WIN_ADVANCED_PERMISSIONS (WIN_ADVANCED_PERMISSION_TYPE+)>
<!ELEMENT WIN_BASIC_PERMISSION_TYPE (#PCDATA)>
<!ELEMENT WIN_ADVANCED_PERMISSION_TYPE (#PCDATA)>

<!ELEMENT SPECIAL (USER, GROUP, DELETION)>
```

```
<!ELEMENT USER (#PCDATA|READ|WRITE|EXECUTE)*>
<!ELEMENT GROUP (#PCDATA|READ|WRITE|EXECUTE)*>
<!ELEMENT OTHER (READ, WRITE, EXECUTE)>
<!ELEMENT DELETION (#PCDATA)>
<!ELEMENT READ (#PCDATA)>
<!ELEMENT WRITE (#PCDATA)>
<!ELEMENT EXECUTE (#PCDATA)>

<!ELEMENT DATA_TYPE (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>
<!ELEMENT TECHNOLOGY_LIST (TECHNOLOGY+)>
<!ELEMENT TECHNOLOGY (ID, NAME, RATIONALE, DATAPOINT?)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT RATIONALE (#PCDATA)>
<!ELEMENT DATAPOINT (CARDINALITY, OPERATOR, DEFAULT_VALUES)>
<!ELEMENT CARDINALITY (#PCDATA)>
<!ELEMENT OPERATOR (#PCDATA)>
<!ELEMENT DEFAULT_VALUES (DEFAULT_VALUE+)>
<!ATTLIST DEFAULT_VALUES total CDATA "0">
<!ELEMENT DEFAULT_VALUE (#PCDATA)>
<!ELEMENT FRAMEWORK_LIST (FRAMEWORK+)>
<!ELEMENT FRAMEWORK (ID, NAME, REFERENCE_LIST)>
<!ELEMENT REFERENCE_LIST (REFERENCE+)>
<!ELEMENT REFERENCE (SECTION, COMMENTS)>
<!ELEMENT SECTION (#PCDATA)>
<!ELEMENT COMMENTS (#PCDATA)>

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

<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!-- EOF -->
```

### Sample - Windows Directory Search Check

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d headers.15
"https://qualysapi.qualys.com/api/2.0/fo/compliance/control/?action=list
&ids=100039"
```

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>2016-09-13T08:43:46Z</DATETIME>
    <CONTROL_LIST>
      <CONTROL>
        <ID>100039</ID>
        <UPDATE_DATE>2016-08-05T05:36:15Z</UPDATE_DATE>
        <CREATED_DATE>2016-08-05T05:36:15Z</CREATED_DATE>
        <CATEGORY>Database Settings</CATEGORY>
        <SUB_CATEGORY><![CDATA[DB Encryption]]></SUB_CATEGORY>
        <STATEMENT><![CDATA[Windows DIR Search check]]></STATEMENT>
        <CRITICALITY>
          <LABEL><![CDATA[UNDEFINED]]></LABEL>
          <VALUE>0</VALUE>
        </CRITICALITY>
        <CHECK_TYPE>
          <![CDATA[Windows Directory Search Check]]></CHECK_TYPE>
        <COMMENT><![CDATA[]]></COMMENT>
        <IGNORE_ERROR>1</IGNORE_ERROR>
        <SCAN_PARAMETERS>
            <BASE_DIR><![CDATA[c:\windows]]></BASE_DIR>
            <DEPTH_LIMIT><![CDATA[3]]></DEPTH_LIMIT>
            <FILE_NAME_MATCH><![CDATA[*.exe]]></FILE_NAME_MATCH>
            <FILE_NAME_SKIP><![CDATA[]]></FILE_NAME_SKIP>
            <DIR_NAME_MATCH>
              <![CDATA[c:\windows\system32]]>
            </DIR_NAME_MATCH>
            <DIR_NAME_SKIP><![CDATA[]]></DIR_NAME_SKIP>
            <WIN_FILE_SYS_OBJECT_TYPES>Directory File
            </WIN_FILE_SYS_OBJECT_TYPES>
            <MATCH_WELL_KNOWN_USERS_FOR_ANY_DOMAIN>
              Yes
            </MATCH_WELL_KNOWN_USERS_FOR_ANY_DOMAIN>
            <WIN_PERMISSION_USERS><![CDATA[AA,AO,LA,BA,Allowed RODC
Password Replication Group,AN,AU,BO,CA,CD,CN,CY,Denied RODC Password
Replication Group,Distributed COM Users,DNS Admins,DNS Update
Proxy,DA,DC,DD,DG,DU,EA,EC,ED,ER,WD,PA,LG,BG,HelpServicesGroup,HA,IIS_IUS
RS,Incoming Forest Trust Builders,IU,SY,NO,NU,LU,MU,PU,RU,PO,RS,ES,RDS
Management Servers,RA,RO,RD,RM,RE,SA,SO,SU,TelnetClients,Terminal Server
License Servers,BU,Windows Authorization Access Group]]>
          </WIN_PERMISSION_USERS>
           <WIN_PERMISSION_MATCH>Any</WIN_PERMISSION_MATCH>
           <WIN_PERMISSIONS>
               <WIN_BASIC_PERMISSIONS>
                   <WIN_BASIC_PERMISSION_TYPE>
                     Full Control
                   </WIN_BASIC_PERMISSION_TYPE>
                   <WIN_BASIC_PERMISSION_TYPE>
                      Modify</WIN_BASIC_PERMISSION_TYPE>
```

```
                        <WIN_BASIC_PERMISSION_TYPE>
                           Read &amp; Execute</WIN_BASIC_PERMISSION_TYPE>
                        <WIN_BASIC_PERMISSION_TYPE>
                           List Folder Content</WIN_BASIC_PERMISSION_TYPE>
                        <WIN_BASIC_PERMISSION_TYPE>
                           Read</WIN_BASIC_PERMISSION_TYPE>
                        <WIN_BASIC_PERMISSION_TYPE>
                           Write</WIN_BASIC_PERMISSION_TYPE>
                        </WIN_BASIC_PERMISSIONS>
                        <WIN_ADVANCED_PERMISSIONS>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Full Control</WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Traverse Folder / Execute Files
                        </WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           List Folder / Read Data
                        </WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Read Attributes</WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Read Extended Attributes
                        </WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Create Files / Write Data
                        </WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Create Folders / Append Data
                        </WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Write Attributes</WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Write Extended Attributes
                        </WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Delete Sub-folders &amp; Files
                        </WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Delete</WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Read Permissions</WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Change Permissions</WIN_ADVANCED_PERMISSION_TYPE>
                        <WIN_ADVANCED_PERMISSION_TYPE>
                           Take Ownership</WIN_ADVANCED_PERMISSION_TYPE>
                    </WIN_ADVANCED_PERMISSIONS>
                </WIN_PERMISSIONS>
                <TIME_LIMIT><![CDATA[300]]></TIME_LIMIT>
                <MATCH_LIMIT><![CDATA[50]]></MATCH_LIMIT>
```

```
            <DATA_TYPE>String List</DATA_TYPE>
            <DESCRIPTION><![CDATA[test]]></DESCRIPTION>
          </SCAN_PARAMETERS>
          <TECHNOLOGY_LIST>
            <TECHNOLOGY>
              <ID>1</ID>
              <NAME>Windows XP desktop</NAME>
              <RATIONALE><![CDATA[test]]></RATIONALE>
              <DATAPOINT>
                  <CARDINALITY>contains</CARDINALITY>
                  <OPERATOR>xeq</OPERATOR>
                  <DEFAULT_VALUES total="1">
                      <DEFAULT_VALUE><![CDATA[1]]></DEFAULT_VALUE>
                  </DEFAULT_VALUES>
              </DATAPOINT>
            </TECHNOLOGY>
            <TECHNOLOGY>
              <ID>2</ID>
              <NAME>Windows 2003 Server</NAME>
              <RATIONALE><![CDATA[test]]></RATIONALE>
              <DATAPOINT>
                  <CARDINALITY>does not contain</CARDINALITY>
                  <OPERATOR>xeq</OPERATOR>
                  <DEFAULT_VALUES total="1">
                      <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
                  </DEFAULT_VALUES>
              </DATAPOINT>
            ...
          </TECHNOLOGY_LIST>
        </CONTROL>
      </CONTROL_LIST>
    </RESPONSE>
</CONTROL_LIST_OUTPUT>
```

### Sample - Unix Directory Search Check

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d headers.15
"https://qualysapi.qualys.com/pi/2.0/fo/compliance/control/?action=list
&ids=100040"
```

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>2016-09-12T10:55:12Z</DATETIME>
<CONTROL_LIST>
  <CONTROL>
    <ID>100040</ID>
    <UPDATE_DATE>2016-08-29T09:36:49Z</UPDATE_DATE>
    <CREATED_DATE>2016-08-05T06:57:55Z</CREATED_DATE>
    <CATEGORY>Anti-Virus/Malware</CATEGORY>
    <SUB_CATEGORY><![CDATA[Virus/Malware Prevention]]></SUB_CATEGORY>
    <STATEMENT><![CDATA[Unix - Directory Search chk]]></STATEMENT>
    <CRITICALITY>
      <LABEL><![CDATA[UNDEFINED]]></LABEL>
      <VALUE>0</VALUE>
    </CRITICALITY>
    <CHECK_TYPE><![CDATA[Unix Directory Search Check]]></CHECK_TYPE>
    <COMMENT><![CDATA[UDC on Unix Directory Search ]]></COMMENT>
    <IGNORE_ERROR>1</IGNORE_ERROR>
    <SCAN_PARAMETERS>
        <BASE_DIR><![CDATA[/usr]]></BASE_DIR>
        <SHOULD_DESCEND><![CDATA[false]]></SHOULD_DESCEND>
        <DEPTH_LIMIT><![CDATA[3]]></DEPTH_LIMIT>
        <FOLLOW_SYMLINK><![CDATA[true]]></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>
        <PERMISSIONS>
            <SPECIAL>
                <USER>any</USER>
                <GROUP>any</GROUP>
                <DELETION>any</DELETION>
            </SPECIAL>
            <USER>
                <READ>any</READ>
                <WRITE>any</WRITE>
                <EXECUTE>any</EXECUTE>
            </USER>
            <GROUP>
                <READ>any</READ>
                <WRITE>any</WRITE>
                <EXECUTE>any</EXECUTE>
            </GROUP>
            <OTHER>
                <READ>any</READ>
                <WRITE>any</WRITE>
                <EXECUTE>any</EXECUTE>
            </OTHER>
        </PERMISSIONS>
        <PERM_COND><![CDATA[all]]></PERM_COND>
        <TYPE_MATCH><![CDATA[d,f,l]]></TYPE_MATCH>
```

```
                <USER_OWNER><![CDATA[Any User]]></USER_OWNER>
                <GROUP_OWNER><![CDATA[Any Group]]></GROUP_OWNER>
                <TIME_LIMIT><![CDATA[300]]></TIME_LIMIT>
                <MATCH_LIMIT><![CDATA[50]]></MATCH_LIMIT>
                <DATA_TYPE>String List</DATA_TYPE>
                <DESCRIPTION><![CDATA[test]]></DESCRIPTION>
              </SCAN_PARAMETERS>
              <TECHNOLOGY_LIST>
                <TECHNOLOGY>
                  <ID>10</ID>
                  <NAME>Solaris 10.x</NAME>
                  <RATIONALE><![CDATA[test]]></RATIONALE>
                  <DATAPOINT>
                      <CARDINALITY>contains</CARDINALITY>
                      <OPERATOR>xre</OPERATOR>
                      <DEFAULT_VALUES total="1">
                          <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
                      </DEFAULT_VALUES>
                  </DATAPOINT>
                </TECHNOLOGY>
                <TECHNOLOGY>
                  <ID>11</ID>
                  <NAME>Red Hat Enterprise Linux 5.x</NAME>
                  <RATIONALE><![CDATA[test]]></RATIONALE>
                  <DATAPOINT>
                      <CARDINALITY>contains</CARDINALITY>
                      <OPERATOR>xre</OPERATOR>
                      <DEFAULT_VALUES total="1">
                          <DEFAULT_VALUE><![CDATA[.*]]></DEFAULT_VALUE>
                      </DEFAULT_VALUES>
                  </DATAPOINT>
                </TECHNOLOGY>
              ...
               </TECHNOLOGY_LIST>
            </CONTROL>
          </CONTROL_LIST>
        </RESPONSE>
      </CONTROL_LIST_OUTPUT>
```

# PC - Changes to STATISTICS element in Policy Report

We will now report statistics information for UDCs in a consistent way using <STATS> under <STATISTICS>. We are already using this format for Windows directory search. Now we'll also use this format for Unix directory search and Windows group membership check.

## Report API

API request (Fetch report):

```
curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWORD" -d
"action=fetch&id=8188" "https://qualysapi.qualys.com/api/2.0/fo/report/"
```

XML output:

```
<COMPLIANCE_POLICY_REPORT>
  <HEADER>
    <NAME>
      <![CDATA[ Policy Report Template ]]>
    </NAME>
    <GENERATION_DATETIME>2016-09-06T04:58:09Z</GENERATION_DATETIME>
    <COMPANY_INFO>
       <NAME>
     ...
     <CHECK>
        <NAME>CHECK1</NAME>
        <DP_NAME>custom.win_group_membership.1001037</DP_NAME>
        <EXPECTED logic="OR">
          <CRITERIA>
            <EVALUATION><![CDATA[contains regular expression
list]]></EVALUATION>
            <V><![CDATA[.*]]></V>
          </CRITERIA>
        </EXPECTED>
        <ACTUAL lastUpdated="2016-08-04T05:20:49Z">
          <V><![CDATA[WIN7-10-10\Administrator]]></V>
          <V><![CDATA[S-1-5-21-2714588763-2906973749-3247541722-
1394]]></V>
          <V><![CDATA[S-1-5-21-2714588763-2906973749-3247541722-
1109]]></V>
        </ACTUAL>
        <STATISTICS>
          <STATS><![CDATA[Group members total: 3]]></STATS>
          <STATS><![CDATA[Group members reported: 3]]></STATS>
```

```
          <STATS><![CDATA[Limit reached: no
      ]]></STATS>
        </STATISTICS>
      </CHECK>
  ...
```

DTD update:

We updated the STATISTICS element to remove ERRORS and add STATS in the Compliance Policy Report DTD (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 STATISTICS (STATS*, SEARCH_DURATION?, ERRORS?)>
<!ELEMENT EVALUATION (#PCDATA)>
...
<!ELEMENT STATS (#PCDATA)>
<!ELEMENT SEARCH_DURATION (#PCDATA)>
<!ELEMENT ERRORS (#PCDATA)>
```

# PC - Last Evaluated Date added to Policy Reports

Your compliance reports (policy report and interactive reports) will now show the date the policy was last evaluated. This date is updated when changes to the policy are saved and when compliance scans on the assets in the policy are processed.

## Report API

API request (Compliance Policy Report):

```
curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWORD" -d
"action=fetch&id=8188" "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>
  ...
    <FILTERS>
      <POLICY><![CDATA[UDCs - policy]]></POLICY>
      <POLICY_LOCKING><![CDATA[Unlocked]]></POLICY_LOCKING>
      <POLICY_LAST_EVALUATED><![CDATA[09/01/2016 at 08:08:37
(GMT)]]></POLICY_LAST_EVALUATED>
    </FILTERS>
  </HEADER>
  <SUMMARY>
    <TOTAL_ASSETS>1</TOTAL_ASSETS>
    <TOTAL_CONTROLS>39</TOTAL_CONTROLS>
    <CONTROL_INSTANCES>
      <TOTAL>22</TOTAL>
      <TOTAL_PASSED>12</TOTAL_PASSED>
      <TOTAL_FAILED>2</TOTAL_FAILED>
      <TOTAL_ERROR>8</TOTAL_ERROR>
      <TOTAL_EXCEPTIONS>0</TOTAL_EXCEPTIONS>
    </CONTROL_INSTANCES>
    <HOST_STATISTICS>
      <HOST_INFO>
        <IP><![CDATA[10.10.10.10]]></IP>
        <TRACKING_METHOD><![CDATA[IP]]></TRACKING_METHOD>
        <DNS><![CDATA[win7-10-10]]></DNS>
        <NETBIOS><![CDATA[WIN7-10-10]]></NETBIOS>
        <OPERATING_SYSTEM><![CDATA[Windows 7 Ultimate 64 bit Edition
Service Pack 1]]></OPERATING_SYSTEM>
        <LAST_SCAN_DATE><![CDATA[2016-08-04T05:20:49Z]]></LAST_SCAN_DATE>
...
```

DTD update:

We added POLICY_LAST_EVALUATED, PC_AGENT_IPS and TRACKING_METHOD to the Compliance Policy Report DTD (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 FILTERS (POLICY, POLICY_LOCKING?, PC_AGENT_IPS?,
POLICY_LAST_EVALUATED)>
<!ELEMENT POLICY (#PCDATA)>
<!ELEMENT POLICY_LOCKING (#PCDATA)>
<!ELEMENT PC_AGENT_IPS (#PCDATA)>
<!ELEMENT POLICY_LAST_EVALUATED (#PCDATA)>
<!ELEMENT SUMMARY (TOTAL_ASSETS, TOTAL_CONTROLS, CONTROL_INSTANCES,
CONTROLS_SUMMARY?, HOST_STATISTICS?)>
...

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

```
<!ELEMENT DP_DESCRIPTIONS (DP*)>
<!ELEMENT DP (DP_NAME, DESCRIPTION, SCAN_PARAMETERS?)>
<!ELEMENT DESCRIPTION (#PCDATA) >

<!ELEMENT SCAN_PARAMETERS (PARAM*)>
<!ELEMENT PARAM (LABEL, VALUE)>
<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>

<!ELEMENT TRACKING_METHOD (#PCDATA)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT DNS (#PCDATA)>
...
<!ELEMENT HOST_STATISTICS (HOST_INFO*)>
<!ELEMENT HOST_INFO (IP, TRACKING_METHOD, DNS, NETBIOS, OPERATING_SYSTEM,
LAST_SCAN_DATE, PERCENTAGE, NETWORK?)>
...
```

**Update for Interactive Reports**

We added POLICY_LAST_EVALUATED to DTDs for interactive reports: Control Pass/Fail Report (control_pass_fail_report.dtd) and Individual Host Report (individual_host_compliance_report.dtd).

XML output for Control Pass/Fail Report (control_pass_fail_report.dtd):

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_PASS_FAIL_REPORT SYSTEM
"https://qualysapi.qualys.com/control_pass_fail_report.dtd">
<CONTROL_PASS_FAIL_REPORT>
  <HEADER>
    <NAME><![CDATA[Control Pass/Fail Report]]></NAME>
    ...
    <USER_INFO>
      <NAME><![CDATA[POC manager]]></NAME>
      <USERNAME>user_name</USERNAME>
      <ROLE>Manager</ROLE>
    </USER_INFO>
    <FILTERS>
      <POLICY><![CDATA[Agent policy]]></POLICY>
      <CID>1048</CID>
      <REFERENCE><![CDATA[]]></REFERENCE>
      <CONTROL><![CDATA[Status of the 'Shutdown: Clear virtual memory
pagefile' setting]]></CONTROL>
      <CRITICALITY>
        <LABEL><![CDATA[CRITICAL]]></LABEL>
        <VALUE>4</VALUE>
      </CRITICALITY>
      <ASSET_GROUP>
```

```
        <TITLE><![CDATA[-]]></TITLE>
      </ASSET_GROUP>
      <ASSET_TAGS>
        <INCLUDED_TAG_SELECTOR><![CDATA[any]]></INCLUDED_TAG_SELECTOR>
        <INCLUDED_TAGS>
          <ASSET_TAG_NAME><![CDATA[Cloud Agent]]></ASSET_TAG_NAME>
        </INCLUDED_TAGS>
      </ASSET_TAGS>
      <DISPLAY><![CDATA[Passed, Failed and Error]]></DISPLAY>
      <SORT_BY><![CDATA[IP Address]]></SORT_BY>
      <POLICY_MODIFIED><![CDATA[09/07/2016 at 10:44:25 (GMT-
0700)]]></POLICY_MODIFIED>
      <POLICY_LAST_EVALUATED><![CDATA[09/07/2016 at 10:44:38 (GMT-
0700)]]></POLICY_LAST_EVALUATED>
    </FILTERS>
  </HEADER>
  <RESULTS>
    <TOTAL_HOSTS>1</TOTAL_HOSTS>
    <TOTAL_FAILED>0</TOTAL_FAILED>
    <TOTAL_PASSED>1</TOTAL_PASSED>
    <PERCENTAGE_PASSED>(100%)</PERCENTAGE_PASSED>
    <TOTAL_ERROR>0</TOTAL_ERROR>
    <HOST_LIST>
      <HOST>
        <TRACKING_METHOD><![CDATA[AGENT]]></TRACKING_METHOD>
        <IP><![CDATA[192.168.248.208]]></IP>
        <DNS><![CDATA[101854-t450]]></DNS>
        <OPERATING_SYSTEM><![CDATA[Microsoft Windows 7 Professional
6.1.7601 Service Pack 1 Build 7601]]></OPERATING_SYSTEM>
        <POSTURE><![CDATA[Passed]]></POSTURE>
      </HOST>
    </HOST_LIST>
  </RESULTS>
</CONTROL_PASS_FAIL_REPORT>
```

DTD update:

```
<!ELEMENT USERNAME (#PCDATA)>
 <!ELEMENT ROLE (#PCDATA)>

<!ELEMENT FILTERS (POLICY, CID, REFERENCE, CONTROL, CRITICALITY?,
ASSET_GROUP, ASSET_TAGS?, DISPLAY, SORT_BY, POLICY_MODIFIED)>
<!ELEMENT FILTERS (POLICY, CID, REFERENCE, CONTROL, CRITICALITY?,
ASSET_GROUP, ASSET_TAGS?, DISPLAY, SORT_BY, POLICY_MODIFIED,
POLICY_LAST_EVALUATED)>
 <!ELEMENT CID (#PCDATA)>
 <!ELEMENT REFERENCE (#PCDATA)>
 <!ELEMENT POLICY (#PCDATA)>
...
```

```
 <!ELEMENT DISPLAY (#PCDATA)>
 <!ELEMENT SORT_BY (#PCDATA)>
 <!ELEMENT POLICY_MODIFIED (#PCDATA)>
<!ELEMENT POLICY_LAST_EVALUATED (#PCDATA)>

 <!ELEMENT ASSET_GROUP (TITLE, TOTAL_SCANIPS?, TOTAL_MAPDOMAINS?,
TOTAL_USERS?, BUSINESS_IMPACT?, DIVISION?, FUNCTION?, LOCATION?,
CVSS_ENVIRO_CDP?, CVSS_ENVIRO_TD?, CVSS_ENVIRO_CR?, CVSS_ENVIRO_IR?,
CVSS_ENVIRO_AR?)>
 <!ELEMENT ASSET_TAGS (INCLUDED_TAG_SELECTOR*, INCLUDED_TAGS*,
EXCLUDED_TAG_SELECTOR*, EXCLUDED_TAGS*)>
...
```

XML output for Individual Host Report (individual_host_compliance_report.dtd):

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE INDIVIDUAL_HOST_COMPLIANCE_REPORT SYSTEM
"https://qualysapi.qualys.com/individual_host_compliance_report.dtd">
<INDIVIDUAL_HOST_COMPLIANCE_REPORT>
  <HEADER>
    <NAME><![CDATA[Individual Host Compliance Report]]></NAME>
    <GENERATION_DATETIME>2016-09-07T21:55:16Z</GENERATION_DATETIME>
    ...
    <USER_INFO>
      <NAME><![CDATA[POC manager]]></NAME>
      <USERNAME>user-name</USERNAME>
      <ROLE>Manager</ROLE>
    </USER_INFO>
    <FILTERS>
      <POLICY><![CDATA[Agent policy]]></POLICY>
      <ASSET_GROUP><![CDATA[]]></ASSET_GROUP>
      <ASSET_TAGS>
        <INCLUDED_TAGS>
          <ASSET_TAG_NAME><![CDATA[Cloud Agent]]></ASSET_TAG_NAME>
        </INCLUDED_TAGS>
      </ASSET_TAGS>
      <IP_ADDRESS><![CDATA[192.168.248.208]]></IP_ADDRESS>
      <DISPLAY><![CDATA[Passed, Failed and Error]]></DISPLAY>

<CRITICALITY_FILTER><![CDATA[UNDEFINED,MINIMAL,MEDIUM,SERIOUS,CRITICAL,UR
GENT]]></CRITICALITY_FILTER>
      <SORT_BY><![CDATA[Order]]></SORT_BY>
      <POLICY_MODIFIED><![CDATA[09/07/2016 at 10:44:25 (GMT-
0700)]]></POLICY_MODIFIED>
      <POLICY_LAST_EVALUATED><![CDATA[09/07/2016 at 10:44:38 (GMT-
0700)]]></POLICY_LAST_EVALUATED>
    </FILTERS>
  </HEADER>
  <RESULTS>
```

```
    <TOTAL_CONTROLS>1013</TOTAL_CONTROLS>
    <TOTAL_FAILED>6</TOTAL_FAILED>
    <PERCENTAGE_FAILED>(0.59%)</PERCENTAGE_FAILED>
    <TOTAL_PASSED>1007</TOTAL_PASSED>
    <PERCENTAGE_PASSED>(99.41%)</PERCENTAGE_PASSED>
    <TOTAL_ERROR>0</TOTAL_ERROR>
    <PERCENTAGE_ERROR></PERCENTAGE_ERROR>
    <HOST>
      <TRACKING_METHOD><![CDATA[AGENT]]></TRACKING_METHOD>
      <IP><![CDATA[192.168.248.208]]></IP>
      <DNS><![CDATA[101854-t450]]></DNS>
      <NETBIOS><![CDATA[101854-T450]]></NETBIOS>
     <OPERATING_SYSTEM><![CDATA[Microsoft Windows 7 Professional 6.1.7601
Service Pack 1 Build 7601]]></OPERATING_SYSTEM>
...
```

DTD update:

```
 <!ELEMENT USERNAME (#PCDATA)>
 <!ELEMENT ROLE (#PCDATA)>

 <!ELEMENT FILTERS (POLICY, ASSET_GROUP, ASSET_TAGS?, IP_ADDRESS, DISPLAY,
CRITICALITY_FILTER?, SORT_BY, POLICY_MODIFIED)>
 <!ELEMENT FILTERS (POLICY, ASSET_GROUP, ASSET_TAGS?, IP_ADDRESS, DISPLAY,
CRITICALITY_FILTER?, SORT_BY, POLICY_MODIFIED, POLICY_LAST_EVALUATED)>
 <!ELEMENT POLICY (#PCDATA)>
 <!ELEMENT ASSET_GROUP (#PCDATA)>
 <!ELEMENT IP_ADDRESS (#PCDATA)>
 ...
 <!ELEMENT CRITICALITY_FILTER (#PCDATA)>
 <!ELEMENT SORT_BY (#PCDATA)>
 <!ELEMENT POLICY_MODIFIED (#PCDATA)>
 <!ELEMENT POLICY_LAST_EVALUATED (#PCDATA)>
 <!ELEMENT ASSET_TAGS (INCLUDED_TAG_SELECTOR*, INCLUDED_TAGS*,
EXCLUDED_TAG_SELECTOR*, EXCLUDED_TAGS*) >
 <!ELEMENT INCLUDED_TAG_SELECTOR (#PCDATA)>
 <!ELEMENT INCLUDED_TAGS (ASSET_TAG_NAME*)>
...
```

# PC - Uniquely Identify Data Points using Name and ID

You can now use the new input parameter "include_dp_name=1" in the Compliance Posture Information API (/api/2.0/fo/compliance/posture/info) to show the name and ID for each data point in the XML output. This is useful for uniquely identifying data points.

## Compliance Posture Information API

API request:

curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWORD" -d headers.15 'https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/?action=list&policy_id=15472&details=All&include_dp_name=1'

XML Response:

```
...
 <DPD_LIST>
        <DPD>
          <LABEL>:dp_1</LABEL>
          <ID>136</ID>
          <NAME><![CDATA[secman.system.clearpageonshut]]></NAME>
         <DESC><![CDATA[This Integer value <B>X</B> indicates the current
status of the setting <B>Shutdown: Clear virtual memory pagefile</B> using
the registry key path
<B>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session
Manager\Memory Management\ClearPageFileAtShutdown</B>.  A value of
<B>0</B> indicates the setting is <B>Disabled</B>; a value of <B>1</B>
indicates the setting is <B>Enabled</B>.]]></DESC>
        </DPD>
...
<DPD>
          <LABEL>:dp_3</LABEL>
          <ID>1001035</ID>
          <NAME><![CDATA[custom.win_group_membership.1001035]]></NAME>
          <DESC><![CDATA[IIS_IUSR]]></DESC>
        </DPD>
...
```

DTD update:

We added the ID and NAME elements to the Posture Information List Output DTD (posture_info_list_output.dtd).

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

...
```

```
<!ELEMENT GLOSSARY (USER_LIST?, HOST_LIST, CONTROL_LIST?,
TECHNOLOGY_LIST?, DPD_LIST?, TP_LIST?, FV_LIST?, TM_LIST?)>
...
<!ELEMENT DPD_LIST (DPD+)>
<!ELEMENT DPD (LABEL, ID?, NAME?, DESC)>
<!ELEMENT DESC (#PCDATA)>

<!ELEMENT TP_LIST (TP+)>
<!ELEMENT TP (LABEL, V+)>
...
```