Qualys Cloud Platform v2.x

API Release Notes

Version 2.34

August 13, 2018

Qualys Cloud Suite API gives you many ways to integrate your programs and API calls with Qualys capabilities. You'll find all the details in our user guides, available at the time of release. Just log in to your Qualys account and go to Help > Resources.

What's New

Fetch Docker information through Asset Management API

Continuous Monitoring (CM) Licensing

New XSS Power Mode Option Profile in WAS

New Security Filters in WAF for Cipher Selection in Web Applications

Separate VULNSIGS information in Asset Management API for split manifest

WAF APIs for version 1.0 deprecated

URL to the Qualys API Server

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

Account Location	API Server URL
Qualys US Platform 1	https://qualysapi.qualys.com
Qualys US Platform 2	https://qualysapi.qg2.apps.qualys.com
Qualys US Platform 3	https://qualysapi.qg3.apps.qualys.com
Qualys EU Platform 1	https://qualysapi.qualys.eu
Qualys EU Platform 2	https://qualysapi.qg2.apps.qualys.eu

Account Location	API Server URL
Qualys India Platform 1	https://qualysapi.qg1.apps.qualys.in
Qualys Private Cloud Platform	https://qualysapi.<customer_base_url>

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

Fetch Docker information through Asset Management API

API affected	/qps/rest/2.0/get/am/hostasset /qps/rest/2.0/search/am/hostasset
New or Updated APIs	Updated
DTD or XSD changes	Yes

The Asset Management API now returns docker (container) information for host assets matching the provided criteria.

Sample 1

Here's sample request and output to get host asset information with docker.

API request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/2.0/get/am/hostasset/7727721"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=
"https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <HostAsset>
            <id>7727721</id>
            <name>10.113.198.121</name>
            <created>2018-06-15T11:51:26Z</created>
            <modified>2018-06-15T11:51:26Z</modified>
            <type>HOST</type>
            <tags>
                <list>
                    <TagSimple>
                        <id>8910214</id>
                        <name>SSD27701</name>
                    </TagSimple>
                    <TagSimple>
                        <id>9252992</id>
                        <name>All_data1</name>
                    </TagSimple>
                </list>
            </tags>
            <qwebHostId>707520</qwebHostId>
            <lastVulnScan>2018-06-15T11:48:58Z</lastVulnScan>
```

```
            <os>CentOS Linux 7.2.1511</os>
            <address>10.113.198.121</address>
            <trackingMethod>IP</trackingMethod>
            <openPort>
                <list>
                    <HostAssetOpenPort>
                        <port>8080</port>
                        <protocol>TCP</protocol>
                        <serviceId>1180</serviceId>
                        <serviceName>HyperText Transport
                           Protocol</serviceName>
                    </HostAssetOpenPort>
                </list>
            </openPort>
            <vuln>
                <list>
                    <HostAssetVuln>
                        <qid>45038</qid>
                       <hostInstanceVulnId>151189845</hostInstanceVulnId>
                        <firstFound>2018-06-15T11:48:58Z</firstFound>
                        <lastFound>2018-06-15T11:48:58Z</lastFound>
                    </HostAssetVuln>
                </list>
            </vuln>
            <networkInterface>
                <list>
                    <HostAssetInterface>
                        <type>LOCAL</type>
                        <address>10.113.198.121</address>
                    </HostAssetInterface>
                </list>
            </networkInterface>
            <isDockerHost>true</isDockerHost>
              <dockerInfo>
                <dockerVersion>18.06.0-ce-rc1</dockerVersion>
                <noOfContainers>1</noOfContainers>
                <noOfImages>2</noOfImages>
              </dockerInfo>
        </HostAsset>
    </data>
</ServiceResponse>
```

Sample 2

Here's sample request and output to search host asset information with docker.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-
binary @- "https://qualysapi.qualys.com/qps/rest/2.0/search/am/hostasset"
< file.xml
```

Note: "file.xml" contains the request POST data.

Request POST data: (Contents of file.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
 <filters>
 <Criteria field="id" operator="EQUALS">7727721</Criteria>
 </filters>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=
"https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <HostAsset>
            <id>7727721</id>
            <name>10.113.198.121</name>
            <created>2018-06-15T11:51:26Z</created>
            <modified>2018-06-15T11:51:26Z</modified>
            <type>HOST</type>
            <tags>
                <list>
                    <TagSimple>
                        <id>8910214</id>
                        <name>SSD27701</name>
                    </TagSimple>
                    <TagSimple>
                        <id>9252992</id>
                        <name>All_data1</name>
                    </TagSimple>
                </list>
            </tags>
```

```
            <qwebHostId>707520</qwebHostId>
            <lastVulnScan>2018-06-15T11:48:58Z</lastVulnScan>
            <os>CentOS Linux 7.2.1511</os>
            <address>10.113.198.121</address>
            <trackingMethod>IP</trackingMethod>
            <openPort>
                <list>
                    <HostAssetOpenPort>
                        <port>8080</port>
                        <protocol>TCP</protocol>
                        <serviceId>1180</serviceId>
                        <serviceName>HyperText Transport
                          Protocol</serviceName>
                    </HostAssetOpenPort>
                </list>
            </openPort>
            <vuln>
                <list>
                    <HostAssetVuln>
                        <qid>6</qid>
                       <hostInstanceVulnId>151189838</hostInstanceVulnId>
                        <firstFound>2018-06-15T11:48:58Z</firstFound>
                        <lastFound>2018-06-15T11:48:58Z</lastFound>
                    </HostAssetVuln>
                    <HostAssetVuln>
                        <qid>45038</qid>
                       <hostInstanceVulnId>151189845</hostInstanceVulnId>
                        <firstFound>2018-06-15T11:48:58Z</firstFound>
                        <lastFound>2018-06-15T11:48:58Z</lastFound>
                    </HostAssetVuln>
                </list>
            </vuln>
            <networkInterface>
                <list>
                    <HostAssetInterface>
                        <type>LOCAL</type>
                        <address>10.113.198.121</address>
                    </HostAssetInterface>
                </list>
            </networkInterface>
            <isDockerHost>true</isDockerHost>
              <dockerInfo>
                <dockerVersion>18.06.0-ce-rc1</dockerVersion>
                <noOfContainers>1</noOfContainers>
                <noOfImages>2</noOfImages>
              </dockerInfo>
        </HostAsset>
    </data>
</ServiceResponse>
```

XSD update

https://qualysapi.qualys.com/qps/xsd/2.0/am/hostasset.xsd

New complexType name added: `HostDockerInfo`

```
...
      <complexType name="HostDockerInfo">
        <sequence>
         <element name="dockerVersion" type="string"/>
         <element name="noOfContainers" type="integer"/>
         <element name="noOfImages" type="integer"/>
        </sequence>
      </complexType>
      <complexType name="HostAsset">
       <complexContent>
        <extension base="tns:Asset">
         <sequence>
          ...
          <element name="processor" type="tns:HostAssetProcessorQList"
           minOccurs="0"/>
          <element name="volume" type="tns:HostAssetVolumeQList" minOccurs="0"/>
          <element name="account" type="tns:HostAssetAccountQList"
           minOccurs="0"/>
          <element name="networkInterface" type="tns:HostAssetInterfaceQList"
           minOccurs="0"/>
          <element name="isDockerHost" type="string" minOccurs="0"/>
          <element name="dockerInfo" type="tns:HostDockerInfo" minOccurs="0"/>
         </sequence>
        </extension>
       </complexContent>
      </complexType>
...
```

Continuous Monitoring (CM) Licensing

APIs affected	/qps/rest/1.0/search/cm/alert/ /qps/rest/1.0/get/cm/alert/<id> /qps/rest/1.0/download/cm/alert/?format=<format> /qps/rest/1.0/search/cm/profile/ /qps/rest/1.0/get/cm/profile/<id>
New or Updated APIs	Updated
DTD or XSD changes	No

With this release asset licensing is implemented in the Continuous Monitoring (CM) app, for internal and external assets. This applies to non trial CM customers only. After login to the CM UI, the customer can add asset tags to be used for licensing under the Configuration tab called Licensing Details. This allows the customer to select the asset tags to enforce the licensing.

Important updates:

- Users are restricted to viewing alerts for assets contained in their license, using both the CM UI and API.

- Users can create monitoring profiles for assets contained in their license, using both the CM UI and API.

How it works - When asset tags are configured for your CM license under Configuration > License Details, the CM app shows alerts only for purchased assets in the CM UI and API, as shown in the license details.

What about my existing monitoring profiles? Once licensing is configured we'll process alerts only for IP addresses in your CM license. If you have a monitoring profile containing IPs not in your CM license, the next time you edit the profile you'll be prompted to change the IPs and select only IPs in your CM license.

Sample - Search Alerts

Find a list of alerts in the user's account for IP address 10.10.30.70. This IP address and monitoring profile IP range 10.10.10.1-10.10.31.255 are included in the user's CM license and are listed as a purchased asset in the user's account under Configuration > License Details.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-
binary @- "https://qualysapi.qualys.com/qps/rest/1.0/search/cm/alert/" <
file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<ServiceRequest>
 <filters>
   <Criteria field="ipAddress" operator="EQUALS">10.10.30.70</Criteria>
 </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/
xsd/1.0/cm/alert.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <Alert>
      <id>244402</id>
      <source>REMEDIATION</source>
      <eventType>SSL_NEW</eventType>
      <triggerUuid>3d41baf9-7caa-4269-9889-d7377aeaace5</triggerUuid>
      <ipAddress>10.10.30.70</ipAddress>
      <hostname>2k3-sp2-25-69.qualys.com</hostname>
      <eventDate>2018-06-04T10:57:43Z</eventDate>
      <alertDate>2018-06-04T10:57:48Z</alertDate>
      <isHidden>true</isHidden>
      <profile>
        <id>7401</id>
        <title>All Critical</title>
        <dateCreated>2017-09-16T19:54:48Z</dateCreated>
        <dateUpdated>2017-09-16T19:54:48Z</dateUpdated>
        <frequency>FREQ_NEVER</frequency>
        <isActive>true</isActive>
        <includedIps>10.10.10.1-10.10.31.255</includedIps>
        <targetList>10.10.10.1-10.10.31.255</targetList>
      </profile>
      <alertInfo>
        <port>0</port>
        <sslName>2k3-sp2-25-69.qualys.com</sslName>
        <sslIssuer>2k3-sp2-25-69.qualys.com</sslIssuer>
      </alertInfo>
    </Alert>
  </data>
</ServiceResponse>
```

Sample - Get Alert Details

View details for alert ID 246213. Alert IP address 10.10.30.240 and the monitoring profile IP range 10.10.10.1-10.10.31.255 are included in the user's CM license and listed as a purchased assets in the user's account under Configuration > License Details.

API request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/1.0/get/cm/alert/246213"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/
xsd/1.0/cm/alert.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Alert>
      <id>246213</id>
      <source>REMEDIATION</source>
      <eventType>HOST_UPDATED</eventType>
      <triggerUuid>3d41baf9-7caa-4269-9889-d7377aeaace5</triggerUuid>
      <ipAddress>10.10.30.240</ipAddress>
      <hostname>win12-30-240</hostname>
      <eventDate>2018-06-04T18:11:54Z</eventDate>
      <alertDate>2018-06-04T18:11:59Z</alertDate>
      <isHidden>false</isHidden>
      <profile>
        <id>7401</id>
        <title>All Critical</title>
        <dateCreated>2017-09-16T19:54:48Z</dateCreated>
        <dateUpdated>2017-09-16T19:54:48Z</dateUpdated>
        <frequency>FREQ_NEVER</frequency>
        <isActive>true</isActive>
        <includedIps>10.10.10.1-10.10.31.255</includedIps>
        <targetList>10.10.10.1-10.10.31.255</targetList>
      </profile>
      <alertInfo>
        <operatingSystem>Windows Server 2012 Standard 64 bit
Edition</operatingSystem>
        <port>0</port>
      </alertInfo>
    </Alert>
  </data>
</ServiceResponse>
```

Sample - Get Profile Details

View details for profile ID 7401. IPs in monitoring profile IP range 10.10.10.1-10.10.31.255 are included in the user's CM license and listed as a purchased assets in the user's account under Configuration > License Details.

API request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/1.0/get/cm/profile/7401"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/
xsd/1.0/cm/profile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Profile>
      <id>7401</id>
      <title>All Critical</title>
      <uuid>d7af450c-828c-4101-a653-737f10d596c6</uuid>
      <dateCreated>2018-06-16T19:54:48Z</dateCreated>
      <dateUpdated>2018-06-16T19:54:48Z</dateUpdated>
      <frequency>FREQ_NEVER</frequency>
      <isActive>true</isActive>
      <includedIps>10.10.10.1-10.10.31.255</includedIps>
      <targetList>10.10.10.1-10.10.31.255</targetList>
      <ruleset>
        <id>4001</id>
        <title>All Critical</title>
        <description>Critical security risks to be addressed
immediately.</description>
        <dateCreated>2018-06-16T19:36:10Z</dateCreated>
        <dateUpdated>2018-06-16T19:36:10Z</dateUpdated>
        <isTemplate>false</isTemplate>
      </ruleset>
    </Profile>
  </data>
</ServiceResponse>
```

New XSS Power Mode Option Profile in WAS

APIs affected	/qps/rest/3.0/get/was/optionprofile/<id> /qps/rest/3.0/create/was/optionprofile /qps/rest/3.0/update/was/optionprofile/<id>
New or Updated APIs	Updated
DTD or XSD changes	Yes

You can now execute specialized scan that performs comprehensive tests for cross-site scripting vulnerabilities using the new option profile with XSS Power Mode detection scope that we have introduced. The detection scope performs tests using the standard XSS payloads, which detect the most common instances of XSS, but also with additional payloads that can identify XSS in certain, less-common situations. Running a scan with option profile that has XSS Power Mode detection scope will provide the best assurance that your web application is free from XSS vulnerabilities.

To launch a scan in the XSS power mode, you need to set the `<xssPowerMode>` element to true under `<detection>` element.

Note: The includedSearchLists/excludeSearchLists, detectionCategories, xssPowerMode elements are mutually exclusive elements. Thus, you can set only one of the elements under detection element.

Sample - Create an option profile with XSS Power Mode detection scope

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-
binary @- "https://qualysapi.qualys.com/qps/rest/3.0/create/was/
optionprofile" < file.xml
```

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

Request POST Data:

```
<ServiceRequest>
   <data>
      <OptionProfile>
         <name>Sample Option Profile With XSS</name>
         <detection>
            <xssPowerMode>true</xssPowerMode>
         </detection>
      </OptionProfile>
   </data>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/w
as/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>1045129</id>
            <name>
                <![CDATA[Launch XSS Power Mode Scan]]>
            </name>
            <owner>
                <id>412791</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>300</maxCrawlRequests>
            <timeoutErrorThreshold>100</timeoutErrorThreshold>
            <unexpectedErrorThreshold>300</unexpectedErrorThreshold>
            <parameterSet>
                <id>0</id>
                <name>
                    <![CDATA[Initial Parameters]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>false</ignoreBinaryFiles>
            <includeActionUriInFormId>false</includeActionUriInFormId>
            <smartScanSupport>false</smartScanSupport>
            <performance>LOW</performance>
            <bruteforceOption>MINIMAL</bruteforceOption>
            <detection>
                <xssPowerMode>true</xssPowerMode>
            </detection>
            <comments>
                <count>0</count>
            </comments>
            <sensitiveContent>
                <creditCardNumber>false</creditCardNumber>
                <socialSecurityNumber>false</socialSecurityNumber>
```

```
            </sensitiveContent>
            <createdDate>2018-07-25T03:45:12Z</createdDate>
            <createdBy>
            <owner>
                <id>412791</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            </createdBy>
            <updatedDate>2018-07-25T03:45:12Z</updatedDate>
            <updatedBy>
             <owner>
                <id>412791</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            </updatedBy>
        </OptionProfile>
    </data>
</ServiceResponse>
```

XSD update

https://qualysapi.qualys.com/qps/xsd/3.0/was/optionprofile.xsd

We have added the new element `xssPowerMode`.

```
...
<xs:element name="detection" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element name="xssPowerMode" type="xs:boolean" minOccurs="0"/>
<xs:element name="detectionCategories" type="DetectionCategoryList"
minOccurs="0"/>
<xs:element name="includedSearchLists" type="SearchListlist"
minOccurs="0"/>
<xs:element name="excludedSearchLists" type="SearchListlist"
minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
...
```

New Security Filters in WAF for Cipher Selection in Web Applications

API affected	/qps/rest/2.0/get/waf/webapp/<id> /qps/rest/2.0/search/waf/webapp/ /qps/rest/2.0/create/waf/webapp /qps/rest/2.0/update/waf/webapp/<id>
New or Updated APIs	Updated
DTD or XSD changes	Yes

We have made cipher selection for your web applications simple with new security filters. You can choose one or more one security filters based on your security requirements. Available security filters are Strong, Good, Weak and Unsafe.

Ciphers are now categorized based on these security filters. We recommend you to choose Strong and Good security filters and corresponding ciphers belonging to these categories for safe and secure communication between your web applications and client browsers.

Input Parameters

New input parameter is described below.

Parameter	Description
sslSecurityFilters	(Text) A comma separated list of allowed SSL security filters (Strong, Good, Weak, Unsafe) Default security filters are Strong, Good. See Security Filters and Corresponding Ciphers.

Sample: List web applications

Here's a sample output showing security filters.

API request:

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
https://qualysapi.qualys.com/qps/rest/2.0/get/waf/webapp/63098273
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/w
af/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
```

```
    <data>
        <WebApp>
            <id>63098273</id>
            <uuid>01bd1b58-2802-48dd-b5b5-ea1342aea21a</uuid>
            ...
            <sslProtocols>
                <![CDATA[SSLV3,TLS10,TLS11,TLS12]]>
            </sslProtocols>
            <sslSecurityFilters>
                <![CDATA[Strong,Good]]>
            </sslSecurityFilters>
            <sslCiphers>
                <![CDATA[ECDHE-RSA-AES256-SHA384,ECDHE-RSA-AES256-GCM-
SHA384,ECDHE-RSA-AES128-SHA256,ECDHE-RSA-AES128-GCM-SHA256,ECDHE-ECDSA-
AES256-SHA384,ECDHE-ECDSA-AES256-GCM-SHA384,ECDHE-ECDSA-AES128-
SHA256,ECDHE-ECDSA-AES128-GCM-SHA256...]]>
            </sslCiphers>
            ....
            <status>DOWN</status>
            <sslEnabled>true</sslEnabled>
            <sslStatus>OK</sslStatus>
            <deploymentStatus>FAILURE</deploymentStatus>
            <deployed>2017-05-31T12:15:14Z</deployed>
        </WebApp>
    </data>
</ServiceResponse>
```

Sample: Create web applications

Here's a sample create request with Strong and Weak security filters.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/create/waf/webapp" <  file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
   <data>
      <WebApp>
         <name>Sslsite1by post</name>
         <url>https://sslsite1.com</url>
         <webServer>
            <id>83002</id>
         </webServer>
```

```
            <webServerTimeout>3600</webServerTimeout>
            <healthcheck>
              <id>122004</id>
              </healthcheck>
              <failureResponseCode>503</failureResponseCode>
              <certificate>
                   <id>92401</id>
               </certificate>
               <sslProtocols>TLS12</sslProtocols>
               <sslSecurityFilters>Strong,Weak</sslSecurityFilters>
             <sslCiphers> ADH-AES128-GCM-SHA256,ADH-AES128-SHA256,ECDHE-RSA-
AES256-SHA384</sslCiphers>
            <blockingMode>true</blockingMode>
            <securityPolicy>
               <id>148003</id>
            </securityPolicy>
            <httpProfile>
               <id>48001</id>
            </httpProfile>
            <clusters>
                   <Cluster>
                       <id>153801</id>
                   </Cluster>
             </clusters>
         </WebApp>
      </data>
   </ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/w
af/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>7831329</id>
            <uuid>ed13870b-66c6-4ba8-8cdd-66aea6c20c36</uuid>
            <name>
                <![CDATA[Sslsite1by post]]>
            </name>
            ...
            <certificate>
                <id>92401</id>
                <uuid>67a52056-dd4f-4644-bbdb-961e5960eebe</uuid>
                <name>
                    <![CDATA[ssl2.33]]>
```

```
                    </name>
                </certificate>
                <sslProtocols>
                    <![CDATA[TLS12]]>
                </sslProtocols>
                <sslSecurityFilters>
                    <![CDATA[Strong,Weak]]>
                </sslSecurityFilters>
                <sslCiphers>
                  <![CDATA[ADH-AES128-GCM-SHA256,ADH-AES128-SHA256,ECDHE-RSA-
AES256-SHA384]]>
                </sslCiphers>
                <blockingMode>true</blockingMode>
                <securityPolicy>
                    <id>148003</id>
                    <uuid>f99cdce6-0c1e-4814-8374-5e1595c9d7c1</uuid>
                    <name>
                        <![CDATA[Copy of portal2.30Sanity]]>
                    </name>
                </securityPolicy>
                ...
            </WebApp>
        </data>
</ServiceResponse>
```

Sample: Error updating web application due to mismatch in security filter and ciphers

Here's a sample error response for choosing incorrect ciphers for Good security filter.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/2.0/update/waf/webapp" <  file.xml
```

Note: "file.xml" contains the request POST data.

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
   <data>
      <WebApp>
         <sslProtocols>TLS11,TLS12</sslProtocols>
         <sslSecurityFilters>Good</sslSecurityFilters>
         <sslCiphers>SRP-RSA-AES-256-CBC-SHA</sslCiphers>
      </WebApp>
   </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/waf/webapp.xsd">
<responseCode>INVALID_PARAM</responseCode>
<responseErrorDetails>
<errorMessage>SSL cipher [SRP-RSA-AES-256-CBC-SHA] is not allowed with the selected security filters.</errorMessage>
</responseErrorDetails>
</ServiceResponse>
```

XSD update

Changes in webapp.xsd (qps/xsd/2.0/waf/webapp.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
    <!-- REQUEST -->
    ...
            <xs:element name="sslProtocols" type="Cdata" minOccurs="0" />
            <xs:element name="sslSecurityFilters" type="Cdata"
minOccurs="0" />
...
                        <xs:element name="deployed" type="xs:dateTime"
minOccurs="0" />
            <xs:element name="synced" type="xs:dateTime" minOccurs="0" />
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Security Filters and Corresponding Ciphers

Strong

Best and highly recommended

ECDHE-RSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES256-GCM-SHA384, ECDH-ECDSA-AES256-GCM-SHA384, ECDH-RSA-AES256-GCM-SHA384, DHE-RSA-AES256-GCM-SHA384

Really Good

ECDHE-ECDSA-AES256-SHA384, ECDHE-RSA-AES256-SHA384, ECDH-RSA-AES256-SHA384, ECDH-ECDSA-AES256-SHA384

Good

Good

DHE-RSA-AES256-SHA256, DH-RSA-AES256-SHA256

Good but with only 128 key

ECDHE-RSA-AES128-GCM-SHA256, ECDHE-ECDSA-AES128-GCM-SHA256, ECDH-RSA-AES128-GCM-SHA256, ECDH-ECDSA-AES128-GCM-SHA256, DHE-RSA-AES128-GCM-SHA256, DH-RSA-AES128-GCM-SHA256

ECDHE-RSA-AES128-SHA256, ECDHE-ECDSA-AES128-SHA256, ECDH-RSA-AES128-SHA256, ECDH-ECDSA-AES128-SHA256, DHE-RSA-AES128-SHA256, DH-RSA-AES128-SHA256

Weak

Not recommended because of high performances consumption issues

DH-RSA-AES256-GCM-SHA384, DH-DS-AES256-GCM-SHA384

Not recommended because of key exchange method

AES256-GCM-SHA384, AES256-SHA256, AES128-SHA256, AES128-GCM-SHA256, ADH-AES128-GCM-SHA256, ADH-AES128-SHA256, ADH-AES256-GCM-SHA384, ADH-AES256-SHA256

Unsafe

These ciphers can be compromised and are not recommended for your web applications.

DSS

DH-DSS-AES128-GCM-SHA256, DH-DSS-AES128-SHA, DH-DSS-AES128-SHA256, DH-DSS-AES256-GCM-SHA384, DH-DSS-AES256-SHA, DH-DSS-AES256-SHA256, DH-DSS-CAMELLIA128-SHA, DH-DSS-CAMELLIA256-SHA, DH-DSS-SEED-SHA, SRP-DSS-AES-128-CBC-SHA, SRP-DSS-AES-256-CBC-SHA, DHE-DSS-AES128-GCM-SHA256, DHE-DSS-AES128-

SHA, DHE-DSS-AES128-SHA256, DHE-DSS-AES256-GCM-SHA384, DHE-DSS-AES256-SHA, DHE-DSS-AES256-SHA256, DHE-DSS-CAMELLIA128-SHA, DHE-DSS-CAMELLIA256-SHA, DHE-DSS-SEED-SHA, DH-DSS-DES-CBC-SHA, DH-DSS-DES-CBC3-SHA, EDH-DSS-DES-CBC-SHA, EDH-DSS-DES-CBC3-SHA, SRP-DSS-3DES-EDE-CBC-SHA

DES

DH-RSA-DES-CBC-SHA, DH-RSA-DES-CBC3-SHA, ECDH-ECDSA-DES-CBC3-SHA, ECDH-RSA-DES-CBC3-SHA, ECDHE-ECDSA-DES-CBC3-SHA, ECDHE-RSA-DES-CBC3-SHA, DES-CBC-SHA, DES-CBC3-SHA, EDH-RSA-DES-CBC-SHA, EDH-RSA-DES-CBC3-SHA

3DES

PSK-3DES-EDE-CBC-SHA, SRP-3DES-EDE-CBC-SHA, SRP-RSA-3DES-EDE-CBC-SHA

RC4

ECDH-ECDSA-RC4-SHA, ECDH-RSA-RC4-SHA, ECDHE-ECDSA-RC4-SHA, ECDHE-RSA-RC4-SHA, RC4-SHA, PSK-RC4-SHA

MD5

RC4-MD5

SHA

AES128-SHA, AES256-SHA, AMELLIA128-SHA, CAMELLIA256-SHA, DH-RSA-AES128-SHA, DH-RSA-CAMELLIA128-SHA, DH-RSA-CAMELLIA256-SHA, DH-RSA-SEED-SHADHE-RSA-AES128-SHA, DHE-RSA-AES256-SHA, DHE-RSA-CAMELLIA128-SHA, DHE-RSA-CAMELLIA256-SHA, DHE-RSA-SEED-SHA, ECDH-ECDSA-AES128-SHA, ECDH-ECDSA-AES256-SHA, ECDH-RSA-AES128-SHA, ECDH-RSA-AES256-SHA, ECDHE-ECDSA-AES128-SHA, ECDHE-ECDSA-AES256-SHA, ECDHE-RSA-AES128-SHA, ECDHE-RSA-AES256-SHA, PSK-AES128-CBC-SHA, PSK-AES256-CBC-SHA, SEED-SHA, SRP-AES-128-CBC-SHA, SRP-AES-256-CBC-SHA, SRP-RSA-AES-128-CBC-SHA, SRP-RSA-AES-256-CBC-SHA

NULL

NULL-SHA256

Separate VULNSIGS information in Asset Management API for split manifest

API affected	/qps/rest/2.0/get/am/hostasset /qps/rest/2.0/search/am/hostasset
New or Updated APIs	Updated
DTD or XSD changes	Yes

The Asset Management API now returns separate VULNSIGS information for host asset when using a split manifest for VM, PC, or SCA.

Sample

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-
binary @- "https://qualysapi.qualys.com/qps/rest/2.0/search/am/hostasset"
< file.xml
```

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

Request POST data: (Contents of file.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
 <filters>
 <Criteria field="id" operator="EQUALS">7866685</Criteria>
 </filters>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.0/a
m/hostasset.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <HostAsset>
            <id>7866685</id>
            <name>ip-172-31-3-82.ap-south-1.compute.internal</name>
            <created>2018-08-01T09:34:44Z</created>
            <modified>2018-08-10T08:39:49Z</modified>
            <type>HOST</type>
            <tags>
```

```
                <list>
                    <TagSimple>
                        <id>10125654</id>
                        <name>Cloud Agent</name>
                    </TagSimple>
                </list>
            </tags>
            <sourceInfo>
                <list>
                    <AssetSource/>
                    <Ec2AssetSourceSimple>
                        <assetId>7866685</assetId>
                        <type>EC_2</type>
                        <firstDiscovered>2018-08-
01T09:34:45Z</firstDiscovered>
                        <lastUpdated>2018-08-01T09:34:45Z</lastUpdated>
                       <reservationId>r-0cd44450f874d4a08</reservationId>
                        <availabilityZone>ap-south-1b</availabilityZone>
                        <privateDnsName>ip-172-31-3-82.ap-south-
1.compute.internal</privateDnsName>
                        <publicDnsName>ec2-13-232-170-59.ap-south-
1.compute.amazonaws.com</publicDnsName>
                        <localHostname>ip-172-31-3-82.ap-south-
1.compute.internal</localHostname>
                        <instanceId>i-0ce729520a8a7d696</instanceId>
                        <instanceType>t2.micro</instanceType>
                        <instanceState>RUNNING</instanceState>
                        <groupId>sg-608b270a</groupId>
                        <groupName>launch-wizard-4</groupName>
                        <spotInstance>false</spotInstance>
                        <accountId>383031258652</accountId>
                        <subnetId>subnet-5a0d6a17</subnetId>
                        <vpcId>vpc-39ccea50</vpcId>
                        <region>ap-south-1</region>
                        <zone>VPC</zone>
                        <imageId>ami-5b673c34</imageId>
                        <publicIpAddress>13.232.170.59</publicIpAddress>
                        <privateIpAddress>172.31.3.82</privateIpAddress>
                        <macAddress>0a:da:e8:58:09:fe</macAddress>
                        <monitoringEnabled>false</monitoringEnabled>
                    </Ec2AssetSourceSimple>
                </list>
            </sourceInfo>
            <qwebHostId>753424</qwebHostId>
            <lastComplianceScan>2018-08-10T00:25:12Z</lastComplianceScan>
            <lastVulnScan>2018-08-10T04:55:06Z</lastVulnScan>
            <lastSystemBoot>2018-08-01T09:23:42Z</lastSystemBoot>
            <lastLoggedOnUser>ec2-user</lastLoggedOnUser>
            <os>Red Hat Enterprise Linux Server 7.5</os>
```

```
            <dnsHostName>ip-172-31-3-82.ap-south-
1.compute.internal</dnsHostName>
            <agentInfo>
                <agentVersion>1.7.1.38</agentVersion>
                <agentId>66fb864e-9609-4324-8eec-48ab6cb7f260</agentId>
                <status>STATUS_ACTIVE</status>
                <lastCheckedIn>2018-08-10T08:39:42Z</lastCheckedIn>
                <connectedFrom>13.232.170.59</connectedFrom>
                <location>Mumbai,Maharashtra India</location>
                <locationGeoLatitude>18.975</locationGeoLatitude>
                <locationGeoLongtitude>72.8258</locationGeoLongtitude>
                <chirpStatus>Inventory Scan Complete</chirpStatus>
                <platform>Linux</platform>
                <activatedModule>AGENT_VM</activatedModule>
                <manifestVersion>
                    <vm>VULNSIGS-VM-0.12.1.0-17</vm>
                    <pc>VULNSIGS-PC-0.17.0.0-27</pc>
                </manifestVersion>
                <agentConfiguration>
                    <id>514001</id>
                    <name>My Default</name>
                </agentConfiguration>
                <activationKey>
                    <activationId>f9391862-de71-4106-9478-
ca14042980dd</activationId>
                    <title>AWS</title>
                </activationKey>
            </agentInfo>
            <networkGuid>6b48277c-0742-61c1-82bb-
cac0f9c4094a</networkGuid>
            <address>13.232.170.59</address>
            <trackingMethod>QAGENT</trackingMethod>
            <totalMemory>990</totalMemory>
            <timezone>UTC</timezone>
            <openPort>
                <list>
                    <HostAssetOpenPort>
                        <port>323</port>
                        <protocol>UDP</protocol>
                    </HostAssetOpenPort>
...
                </list>
            </openPort>
            <software>
                <list>
                    <HostAssetSoftware>
                        <name>GeoIP</name>
                        <version>1.5.0-11.el7</version>
                    </HostAssetSoftware>
```

```
        <HostAssetSoftware>
            <name>NetworkManager</name>
            <version>1.10.2-13.el7</version>
        </HostAssetSoftware>
        ...
    </list>
</software>
<vuln>
    <list>
        <HostAssetVuln>
            <qid>370198</qid>
           <hostInstanceVulnId>157377851</hostInstanceVulnId>
            <firstFound>2018-08-06T10:08:37Z</firstFound>
            <lastFound>2018-08-10T04:55:06Z</lastFound>
        </HostAssetVuln>
        <HostAssetVuln>
            <qid>370472</qid>
           <hostInstanceVulnId>157377852</hostInstanceVulnId>
            <firstFound>2018-08-06T10:08:37Z</firstFound>
            <lastFound>2018-08-10T04:55:06Z</lastFound>
        </HostAssetVuln>
        ...
    </list>
</vuln>
<processor>
    <list>
        <HostAssetProcessor>
            <name>Intel(R) Xeon(R)</name>
            <speed>2400</speed>
        </HostAssetProcessor>
    </list>
</processor>
<volume>
    <list>
        <HostAssetVolume>
            <name>/</name>
            <size>10724814848</size>
            <free>9259859968</free>
        </HostAssetVolume>
        ...
    </list>
</volume>
<account>
    <list>
        <HostAssetAccount>
            <username>root</username>
        </HostAssetAccount>
        <HostAssetAccount>
            <username>ec2-user</username>
```

```
                        </HostAssetAccount>
                    </list>
                </account>
                <networkInterface>
                    <list>
                        <HostAssetInterface>
                            <interfaceName>eth0</interfaceName>
                            <macAddress>0a:da:e8:58:09:fe</macAddress>
                            <type>LOCAL</type>
                            <address>fe80:0:0:0:8da:e8ff:fe58:9fe</address>
                            <gatewayAddress>172.31.0.1</gatewayAddress>
                        </HostAssetInterface>
                        ...
                    </list>
                </networkInterface>
            </HostAsset>
        </data>
    </ServiceResponse>
```

XSD update

https://qualysapi.qualys.com/qps/xsd/2.0/am/agent_source.xsd

We have added new complexType with name `ManifestVersion`.

```
    ...
    <complexType name="QAgentAssetSource">
       <sequence>
        ...
        <element name="localIpv4" type="string" minOccurs="0"/>
        <element name="localIpv6" type="string" minOccurs="0"/>
        <element name="manifestVersion" type="tns:ManifestVersion"
         minOccurs="0"/>
        <element name="AgentConfiguration" type="tns:AgentConfig"
         minOccurs="0"/>
       <element name="ActivationKey" type="tns:ActivationKey" minOccurs="0"/>
       </sequence>
    </complexType>
     <!--  DATA  -->
    ...
       <complexType name="ManifestVersion">
         <sequence>
          <element name="vm" type="string" minOccurs="0"/>
          <element name="pc" type="string" minOccurs="0"/>
          <element name="sca" type="string" minOccurs="0"/>
         </sequence>
       </complexType>
...
```

WAF APIs for version 1.0 deprecated

WAF APIs for version 1.0 are now deprecated and no longer available. You can use equivalent version 2.0 APIs to perform WAF operations.

Here is a list of 1.0 APIs which are removed:

Web Applications

/qps/rest/1.0/get/waf/webapp/

/qps/rest/1.0/count/waf/webapp/

/qps/rest/1.0/search/waf/webapp/

/qps/rest/1.0/create/waf/webapp/

/qps/rest/1.0/update/waf/webapp/

/qps/rest/1.0/delete/waf/webapp/

WAF Cluster

/qps/rest/1.0/get/waf/cluster/

/qps/rest/1.0/count/waf/cluster/

/qps/rest/1.0/search/waf/cluster/

/qps/rest/1.0/create/waf/cluster/

/qps/rest/1.0/update/waf/cluster/

/qps/rest/1.0/delete/waf/cluster/

WAF Appliance

/qps/rest/1.0/get/waf/appliance/

/qps/rest/1.0/count/waf/appliance/

/qps/rest/1.0/search/waf/appliance/

/qps/rest/1.0/delete/waf/appliance/

Refer to Qualys Web Application Firewall API User Guide for detailed information on using version 2.0 APIs.