Qualys Cloud Platform v2.x

API Release Notes

Version 2.44

February 3, 2020

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

WAS API: SSL Lab Information Support in WAS

WAS API: Notification for Huge Reports

URL to the Qualys API Server

The Qualys API URL you should use for API requests depends on the Qualys platform where your account is located.

Click here to identify your Qualys platform and get the API URL

This documentation uses the API gateway URL for Qualys US Platform 1 (https://gateway.qg1.apps.qualys.com) in sample API requests. If you're on another platform, please replace this URL with the appropriate gateway URL for your account.

WAS API: SSL Lab Information Support in WAS

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

We now detect and report SSL/TLS and Certificate related vulnerabilities in WAS.

Permissions

- The user must have the WAS module enabled.

- The "API Access" and "Access WAS module" permissions must be enabled for the user

- The web application must be within the user's scope.

Sample - Get details of findings with "SSL/TLS and Certificate issues"

Let us fetch details of a finding that includes different types of SSL/TLS and Certificate issues. The Information Gathered type of finding includes the details. The different types of SSL/TLS and certificate issues that we support are:

- SSL Data with Certificate Fingerprint

- SSL Data with Prop

- SSL Data with Kex

- SSL Data with Ciphers

The finding you view could include one or multiple issue types listed above. The name tag indicates the type of the issue.

API request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/finding/581856"
```

XML output (SSL Data with Certificate Fingerprint):

```
<?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/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
```

```
    <data>
        <Finding>
            <id>581856</id>
            <uniqueId>d6a88c61-fcda-4f46-9767-1d8cb521d953</uniqueId>
            <qid>86002</qid>
            <name>
                <![CDATA[SSL Certificate - Information]]>
            </name>
            <type>INFORMATION_GATHERED</type>
            <findingType>QUALYS</findingType
               ...
                <sslDataInfoList>
                    <list>
                        <SSLDataInfo>

<certificateFingerprint>291126AC8ED272F71EDF06E5B76BBECD1C811769D4FE988D
E95FF848AFEBCF6A</certificateFingerprint>
                        </SSLDataInfo>
                    </list>
                </sslDataInfoList>
            </sslData>
        </Finding>
    </data>
</ServiceResponse>
```

API request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/finding/581863"
```

XML output (SSL Data with Prop):

```
<?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/finding.xsd">
<responseCode>SUCCESS</responseCode>
<count>1</count>
    <data>
        <Finding>
            <id>581863</id>
            <uniqueId>b90679ec-96ec-4ffc-b027-011924dff64a</uniqueId>
            <qid>38706</qid>
            <name>
                <![CDATA[SSL/TLS Protocol Properties]]>
            </name>
            <type>INFORMATION_GATHERED</type>
            <findingType>QUALYS</findingType>
            ...
```

```
                    <sslDataInfoList>
                        <list>
                            <SSLDataInfo>
                                <sslDataPropList>
                                    <list>
                                        <SSLDataProp>
                                          <name>Extended Master Secret</name>
                                            <value>yes</value>
                                            <protocol>TLSv1</protocol>
                                        </SSLDataProp>
                                        <SSLDataProp>
                                            <name>Encrypt Then MAC</name>
                                            <value>yes</value>
                                            <protocol>TLSv1</protocol>
                                        </SSLDataProp>
                                    </list>
                                </sslDataPropList>
                            </SSLDataInfo>
                        </list>
                    </sslDataInfoList>
                </sslData>
            </Finding>
        </data>
    </ServiceResponse>
```

API request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/finding/581864"
```

XML output (SSL Data with Kex):

```
 <?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/finding.xsd">
<responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>581864</id>
            <uniqueId>9e5f4a6e-9d20-4690-bf89-f54d2d74cb0e</uniqueId>
            <qid>38704</qid>
            <name>
                <![CDATA[SSL/TLS Key Exchange Methods]]>
            </name>
           <type>INFORMATION_GATHERED</type>
            <findingType>QUALYS</findingType>
...
```

```
<sslDataInfoList>
                    <list>
                        <SSLDataInfo>
                            <sslDataKexList>
                                <list>
                                    <SSLDataKex>
                                        <protocol>TLSv1</protocol>
                                        <kex>ECDHE</kex>
                                        <group>x25519</group>
                                        <keysize>256</keysize>
                                        <fwdsec>yes</fwdsec>
                                        <classical>128</classical>
                                        <quantum>low</quantum>
                                    </SSLDataKex>
...
          </sslData>
        </Finding>
    </data>
</ServiceResponse>
```

API request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/finding/581861"
```

XML output (SSL Data with Cipher):

```
 <?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/finding.xsd">
<responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>581861</id>
            <uniqueId>17797d3b-c648-47e0-b172-e6ff7dbf150f</uniqueId>
            <qid>38116</qid>
            <name>
                <![CDATA[SSL Server Information Retrieval]]>
            </name>
            <type>INFORMATION_GATHERED</type>
            <findingType>QUALYS</findingType>
...
 <sslDataInfoList>
                    <list>
                        <SSLDataInfo>
                            <sslDataCipherList>
                                <list>
```

```
                                        <SSLDataCipher>
                                            <protocol>TLSv1</protocol>
                                            <name>ECDHE-RSA-AES128-SHA</name>
                                            <keyExchange>ECDH</keyExchange>
                                            <auth>RSA</auth>
                                            <mac>SHA1</mac>
                                            <encryption>AES(128)</encryption>
                                            <grade>MEDIUM</grade>
                                        </SSLDataCipher>
...
          </sslData>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - Search findings with "SSL/TLS and Certificate issues"

Let us search finding details using the unique ID of a finding.

API request:

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

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

Request POST Data (file.xml):

```
<ServiceRequest>
    <preferences>
        <verbose>true</verbose>
    </preferences>
    <filters>
        <Criteria field="uniqueId" operator="EQUALS">d6a88c61-fcda-4f46-
9767-1d8cb521d953</Criteria>
    </filters>
</ServiceRequest>
```

XML output:

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

<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/3.0/wa
s/finding.xsd">
<responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
```

```
            <id>581856</id>
            <uniqueId>d6a88c61-fcda-4f46-9767-1d8cb521d953</uniqueId>
            <qid>86002</qid>
            <name>
                <![CDATA[SSL Certificate - Information]]>
            </name>
            <type>INFORMATION_GATHERED</type>
            <findingType>QUALYS</findingType>
...
<sslDataInfoList>
                    <list>
                        <SSLDataInfo>

<certificateFingerprint>291126AC8ED272F71EDF06E5B76BBECD1C811769D4FE988DE
95FF848AFEBCF6A</certificateFingerprint>
                        </SSLDataInfo>
                    </list>
                </sslDataInfoList>
            </sslData>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - Create option profile with "SSL/TLS and Certificate issues"

You can execute specialized scan that performs tests for SSL/TLS and Certificate related vulnerabilities using the option profile with SSL/TLS and Certificate category configured in the API request.

API request:

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

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

Request POST Data (file.xml):

```
<ServiceRequest>
 <data>
  <OptionProfile>
   <name><![CDATA[Option Profile with SSL data]]></name>
    <detection>
        <detectionCategories>
            <set>
                <DetectionCategory>
                    <name>SSL/TLS and Certificate issues</name>
                </DetectionCategory>
```

```
            </set>
          </detectionCategories>
      </detection>
    </OptionProfile>
  </data>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/3.0/wa
s/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>897483</id>
            <name>
                <![CDATA[My Option Profile - SSL data]]>
            </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>
            ...
            <detection>
                <detectionCategories>
                    <count>1</count>
                    <list>
                        <DetectionCategory>
                            <id>152</id>
                            <name>SSL/TLS and Certificate issues</name>
                        </DetectionCategory>
                    </list>
                </detectionCategories>
                <enableXssPayloads>false</enableXssPayloads>
            </detection>
            ...
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Update option profile with "SSL/TLS and Certificate issues"

Let us update an existing option profile to include the SSL/TLS and certificate related vulnerabilities.

API request:

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

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

Request POST Data (file.xml):

```
<ServiceRequest>
 <data>
  <OptionProfile>
   <detection>
        <detectionCategories>
            <set>
                <DetectionCategory>
                    <name>SSL/TLS and Certificate issues</name>
                </DetectionCategory>
            </set>
        </detectionCategories>
    </detection>
  </OptionProfile>
 </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/3.0/w
as/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>897483</id>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Get details of option profile with "SSL/TLS and Certificate issues"

Let us view the details of an option profile that has SSL/TLS detection enabled.

API request:

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/optionprofile/832265669"
```

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/3.0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>832265669</id>
        ...
           <detection>
                <detectionCategories>
                    <count>1</count>
                    <list>
                        <DetectionCategory>
                            <id>152</id>
                            <name>SSL/TLS and Certificate issues</name>
                        </DetectionCategory>
                    </list>
                </detectionCategories>
                <enableXssPayloads>false</enableXssPayloads>
            </detection>
      ...
     </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Download Scan Results

Let us view the details of an scan results that has SSL/TLS issue details.

API request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/download/was/report/1302"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8"?>
<WasScan xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/3.0/wa
s/wasscan.xsd">
    <id>3217161</id>
    <name>
        <![CDATA[[[SSL-Certs]] 2020-01-30 6:20:49PM]]>
    </name>
    <reference>was/1580388655076.626241</reference>
    <type>VULNERABILITY</type>
    <mode>ONDEMAND</mode>
    <progressiveScanning>DISABLED</progressiveScanning>
    <multi>false</multi>
    <target>
        <webApp>
            <id>3016632</id>
            <name>
                <![CDATA[SSL-Certs]]>
            </name>
            <url>
                <![CDATA[https://10.115.78.72/welcome.html]]>
            </url>
        </webApp>
        <scannerAppliance>
            <type>INTERNAL</type>
            <friendlyName>
                <![CDATA[WAS_Scanner_vp1]]>
            </friendlyName>
        </scannerAppliance>
        <cancelOption>SPECIFIC</cancelOption>
    </target>
    <profile>
        <id>893488</id>
        <name>
            <![CDATA[ssl]]>
        </name>
    </profile>
    <options>
        <count>16</count>
        <list>
            <WasScanOption>
                <name>Web Application Authentication Record Name</name>
                <value>
                    <![CDATA[None]]>
                </value>
            </WasScanOption>
...
<list>
            <WasScanIg>
                <qid>38704</qid>
                <title>
```

```
                    <![CDATA[SSL/TLS Key Exchange Methods]]>
                </title>
                <sslData>
...
<sslDataInfoList>
                        <list>
                            <SSLDataInfo>
                                <sslDataKexList>
                                    <list>
                                        <SSLDataKex>
                                            <protocol>TLSv1</protocol>
                                            <kex>ECDHE</kex>
                                            <group>x25519</group>
                                            <keysize>256</keysize>
                                            <fwdsec>yes</fwdsec>
                                            <classical>128</classical>
                                            <quantum>low</quantum>
                                        </SSLDataKex>
...
<WasScanIg>
                <qid>38706</qid>
                <title>
                    <![CDATA[SSL/TLS Protocol Properties]]>
                </title>
                <sslData>
...
<sslDataInfoList>
                        <list>
                            <SSLDataInfo>
                                <sslDataPropList>
                                    <list>
                                        <SSLDataProp>
                                            <name>Extended Master
Secret</name>
                                            <value>yes</value>
                                            <protocol>TLSv1</protocol>
                                        </SSLDataProp>
                                        <SSLDataProp>
                                            <name>Encrypt Then MAC</name>
                                            <value>yes</value>
                                            <protocol>TLSv1</protocol>
                                        </SSLDataProp>
...
<WasScanIg>
                <qid>6</qid>
                <title>
                    <![CDATA[DNS Host Name]]>
                </title>
                <sslData>
```

```
...
<sslDataInfoList>
                        <list>
                            <SSLDataInfo>

<certificateFingerprint>291126AC8ED272F71EDF06E5B76BBECD1C811769D4FE988DE
95FF848AFEBCF6A</certificateFingerprint>
                            </SSLDataInfo>
                        </list>
                    </sslDataInfoList>
...
<WasScanIg>
                <qid>38291</qid>
                <title>
                    <![CDATA[SSL Session Caching Information]]>
                </title>
...
<WasScanIg>
                <qid>45017</qid>
                <title>
                    <![CDATA[Operating System Detected]]>
                </title>
                <sslData>
                    <protocol>tcp</protocol>
                    <ip>10.115.78.72</ip>
                    <port>0</port>
                    <result>

<![CDATA[Ubuntu_/_Fedora_/_Tiny_Core_Linux_/_Linux_3.x TCP/IP_Fingerprint
U5933:443
]]>
                    </result>
                </sslData>
...
<WasScanIg>
                <qid>38116</qid>
                <title>
                    <![CDATA[SSL Server Information Retrieval]]>
                </title>
                <sslData>
...
<sslDataInfoList>
                        <list>
                            <SSLDataInfo>
                                <sslDataCipherList>
                                    <list>
                                        <SSLDataCipher>
                                            <protocol>TLSv1</protocol>
                                        <name>ECDHE-RSA-AES128-SHA</name>
```

```
                                                    <keyExchange>ECDH</keyExchange>
                                                      <auth>RSA</auth>
                                                      <mac>SHA1</mac>
                                                   <encryption>AES(128)</encryption>
                                                      <grade>MEDIUM</grade>
                                                   </SSLDataCipher>
...
 </igs>
    <sendMail>true</sendMail>
    <enableWAFAuth>false</enableWAFAuth>
</WasScan>
```

XSD Update

Changes in wasscan.xsd (.../qps/xsd/3.0/was/wasscanxsd).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
...
 <xs:complexType name="SSLData">
        <xs:all>
            <xs:element name="flags" type="xs:string" minOccurs="0" />
            <xs:element name="protocol" type="xs:string" minOccurs="0" />
           <xs:element name="virtualhost" type="xs:string" minOccurs="0"
/>
            <xs:element name="ip" type="xs:string" minOccurs="0" />
            <xs:element name="port" type="xs:string" minOccurs="0" />
            <xs:element name="result" type="Cdata" minOccurs="0" />
            <xs:element name="sslDataInfoList" type="SSLDataInfoList"
minOccurs="0" />
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataInfoList">
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                       <xs:element name="SSLDataInfo" type="SSLDataInfo"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>
```

```
    <xs:complexType name="SSLDataInfo">
        <xs:all>
            <xs:element name="certificateFingerprint" type="xs:string"
minOccurs="0" />
           <xs:element name="sslDataCipherList" type="SSLDataCipherList"
minOccurs="0" />
            <xs:element name="sslDataKexList" type="SSLDataKexList"
minOccurs="0" />
            <xs:element name="sslDataPropList" type="SSLDataPropList"
minOccurs="0" />
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataCipherList">
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="SSLDataCipher"
type="SSLDataCipher" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataKexList">
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="SSLDataKex" type="SSLDataKex"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataPropList">
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                       <xs:element name="SSLDataProp" type="SSLDataProp"
maxOccurs="unbounded"/>
```

```
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataCipher">
        <xs:all>
            <xs:element name="protocol" type="xs:string" minOccurs="0" />
            <xs:element name="name" type="xs:string" minOccurs="0" />
           <xs:element name="keyExchange" type="xs:string" minOccurs="0"
/>
            <xs:element name="auth" type="xs:string" minOccurs="0" />
            <xs:element name="mac" type="xs:string" minOccurs="0" />
          <xs:element name="encryption" type="xs:string" minOccurs="0" />
            <xs:element name="grade" type="xs:string" minOccurs="0" />
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataKex">
        <xs:all>
            <xs:element name="protocol" type="xs:string" minOccurs="0" />
            <xs:element name="kex" type="xs:string" minOccurs="0" />
            <xs:element name="group" type="xs:string" minOccurs="0" />
            <xs:element name="keysize" type="xs:string" minOccurs="0" />
            <xs:element name="fwdsec" type="xs:string" minOccurs="0" />
           <xs:element name="classical" type="xs:string" minOccurs="0" />
            <xs:element name="quantum" type="xs:string" minOccurs="0" />
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataProp">
        <xs:all>
            <xs:element name="name" type="xs:string" minOccurs="0" />
            <xs:element name="value" type="xs:string" minOccurs="0" />
            <xs:element name="protocol" type="xs:string" minOccurs="0" />
        </xs:all>
    </xs:complexType>

</xs:schema>
...
     </xs:all>
    </xs:complexType>
 </xs:schema>
```

Changes in finding.xsd (.../qps/xsd/3.0/was/finding.xsd).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
...
<xs:complexType name="SSLData">
        <xs:all>
            <xs:element name="flags" type="xs:string" minOccurs="0" />
            <xs:element name="protocol" type="xs:string" minOccurs="0" />
           <xs:element name="virtualhost" type="xs:string" minOccurs="0"
/>
            <xs:element name="ip" type="xs:string" minOccurs="0" />
            <xs:element name="port" type="xs:string" minOccurs="0" />
            <xs:element name="result" type="Cdata" minOccurs="0" />
            <xs:element name="sslDataInfoList" type="SSLDataInfoList"
minOccurs="0" />
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataInfoList">
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                       <xs:element name="SSLDataInfo" type="SSLDataInfo"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataInfo">
        <xs:all>
            <xs:element name="certificateFingerprint" type="xs:string"
minOccurs="0" />
           <xs:element name="sslDataCipherList" type="SSLDataCipherList"
minOccurs="0" />
            <xs:element name="sslDataKexList" type="SSLDataKexList"
minOccurs="0" />
            <xs:element name="sslDataPropList" type="SSLDataPropList"
minOccurs="0" />
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataCipherList">
```

```
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="SSLDataCipher"
type="SSLDataCipher" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataKexList">
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="SSLDataKex" type="SSLDataKex"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataPropList">
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                       <xs:element name="SSLDataProp" type="SSLDataProp"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataCipher">
        <xs:all>
            <xs:element name="protocol" type="xs:string" minOccurs="0" />
            <xs:element name="name" type="xs:string" minOccurs="0" />
           <xs:element name="keyExchange" type="xs:string" minOccurs="0"
/>
            <xs:element name="auth" type="xs:string" minOccurs="0" />
            <xs:element name="mac" type="xs:string" minOccurs="0" />
```

```
            <xs:element name="encryption" type="xs:string" minOccurs="0" />
             <xs:element name="grade" type="xs:string" minOccurs="0" />
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataKex">
        <xs:all>
            <xs:element name="protocol" type="xs:string" minOccurs="0" />
            <xs:element name="kex" type="xs:string" minOccurs="0" />
            <xs:element name="group" type="xs:string" minOccurs="0" />
            <xs:element name="keysize" type="xs:string" minOccurs="0" />
            <xs:element name="fwdsec" type="xs:string" minOccurs="0" />
           <xs:element name="classical" type="xs:string" minOccurs="0" />
            <xs:element name="quantum" type="xs:string" minOccurs="0" />
        </xs:all>
    </xs:complexType>

    <xs:complexType name="SSLDataProp">
        <xs:all>
            <xs:element name="name" type="xs:string" minOccurs="0" />
            <xs:element name="value" type="xs:string" minOccurs="0" />
            <xs:element name="protocol" type="xs:string" minOccurs="0" />
        </xs:all>
    </xs:complexType>

</xs:schema>
```

WAS API: Notification for Huge Reports

API affected	/qps/rest/3.0/create/was/report
New or Updated APIs	Updated API
DTD or XSD changes	No

Report creation may sometimes fail if the report is created for large number of web applications or scans. To avoid such failures, we have now categorized report creation as per the number of web applications or scans being included in the report.

The categorization is as follows

Number of Web Applications/Scans	Create Report (API)
Less than or equal to 500	Yes
More than 500	No

For web applications or scans less than or equal to 500, you can create the report. But if the number of web applications or scans exceeds 500, report cannot be created and error message is displayed in such cases.

Permissions

- The user must have the WAS module enabled.

- The "API Access" and "Create Report" permissions must be enabled for the user

- The web application must be within the user's scope.

Input Parameters

No change in input parameters.

Sample - Create Report for more than 500 web applications

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/report" < file.xml
```

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

Request POST Data (file.xml):

```
<ServiceRequest>
```

```
        <data>
            <Report>
                   <name><![CDATA[Web Application Report]]></name>
                    <format>HTML_BASE64</format>
                    <type>WAS_WEBAPP_REPORT</type>
                    <config>
                        <webAppReport>
                           <target>
                             <tags>
                              <included>
                                <option>ALL</option>
                                <tagList>
                                   <Tag>
                                     <id>102148815</id>
                                   </Tag>
                                </tagList>
                              </included>
                            </tags>
                           </target>
                        </webAppReport>
                    </config>
            </Report>
        </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/3.0/w
    as/optionprofile.xsd">
        <responseCode>INVALID_REQUEST</responseCode>
        <responseErrorDetails>
           <errorMessage>The report can not be generated because it has more
    than 500 web applications.</errorMessage>
        </responseErrorDetails>
    </ServiceResponse>
```