



Qualys Cloud Platform v2.x

API Release Notes

Version 2.40
July 16, 2019

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: Option to choose all Detections in Option Profile](#)

[WAS API: Unique ID for Findings](#)

URL to the Qualys API Server

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

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

WAS API: Option to choose all Detections in Option Profile

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

You can now configure the option profile for you scan so that it could include all the WAS related detections. We have now introduced a new option named "Everything" for detection scope of Option Profile that includes every WAS related detection during the scan.

Input Parameters

We have enhanced the detection parameter to include the new element value EVERYTHING for <detectionScope>.

Parameter	Description
detection (keyword)	<p>Select if scans launched with this profile shall perform a full assessment for all WAS detections the engine is able to discover, or if the scan shall focus on the detection of specific vulnerabilities and/or information: Core, Categories, Custom Search list, XSS Power Mode, Everything.</p> <p>If <detectionScope> is present then the detection scope = CORE or EVERYTHING Core: Core scope includes vulnerabilities that Qualys considers most common in today's web applications. It does not include all the vulnerabilities that WAS can detect. Everything: Everything scope includes all the vulnerabilities that WAS can detect.</p> <p>Example:</p> <pre><detection> <detectionScope>EVERYTHING</detectionScope> </detection></pre> <p>Note: The <includedSearchLists>, <excludedSearchLists>, <detectionCategories>, <xssPowerMode>, <detectionScope> elements are mutually exclusive elements.</p>

Sample 1 - Create Option Profile with Everything as 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><![CDATA[Sample Option Profile]]></name>  
      <detection>  
        <detectionScope>EVERYTHING</detectionScope>  
      </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/was/optionprofile.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <OptionProfile>  
      <id>78744</id>  
      <name>  
        <![CDATA[Sample Option Profile]]>  
      </name>  
      <owner>  
        <id>337590</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>
<enhancedCrawling>>false</enhancedCrawling>
<smartScanSupport>>false</smartScanSupport>
<performance>LOW</performance>
<bruteforceOption>MINIMAL</bruteforceOption>
<strong>detection</strong>
  <strong>detectionScope</strong>EVERYTHING</strong>
</strong>
<comments>
  <count>0</count>
</comments>
...
</updatedBy>
</OptionProfile>
</data>
</ServiceResponse>
```

Sample 2 - Update option profile to change detection scope to Everything

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"--data-
binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/83226
5669" < file.xml
```

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

Request POST Data (file.xml):

```
<ServiceRequest>
  <data>
    <OptionProfile>
      <detection>
        <detectionScope>EVERYTHING</detectionScope>
      </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>832265669</id>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Sample 3 - Get details of an option profile to know the detection scope

API request:

```
curl -u "USERNAME:PASSWORD" " -X GET -H "Content-type: text/xml"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/optionprofile/77683"
```

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>77683</id>
      <name>
        <![CDATA[Sample Option Profile]]>
      </name>
      <owner>
        <id>337590</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>1000</maxCrawlRequests>
      <timeoutErrorThreshold>100</timeoutErrorThreshold>
      <unexpectedErrorThreshold>300</unexpectedErrorThreshold>
      <parameterSet>
        <id>0</id>
```

```
        <name>
            <![CDATA[Initial Parameters]]>
        </name>
    </parameterSet>
    <ignoreBinaryFiles>true</ignoreBinaryFiles>
    <includeActionUriInFormId>false</includeActionUriInFormId>
    <enhancedCrawling>false</enhancedCrawling>
    <smartScanSupport>true</smartScanSupport>
    <smartScanDepth>5</smartScanDepth>
    <performance>LOW</performance>
    <bruteforceOption>MINIMAL</bruteforceOption>
    <b>detection</b>
        <b>detectionScope</b>EVERYTHING</b></b>
    </b>
    <comments>
        <count>0</count>
    </comments>
    ...
    </updatedBy>
</OptionProfile>
</data>
</ServiceResponse>
```

XSD Update

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

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  ...
  <xs:element name="bruteforceList" type="BruteforceList" minOccurs="0"/>
  <xs:element name="numberOfAttempts" type="xs:long" minOccurs="0"/>
  <xs:element name="detection" minOccurs="0">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="detectionScope" default="CORE" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="CORE"/>
              <xs:enumeration value="EVERYTHING"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <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>
  ...
  </xs:element>
</xs:all>
</xs:complexType>
</xs:schema>
```

WAS API: Unique ID for Findings

API affected	/qps/rest/3.0/get/was/finding/<id> /qps/rest/3.0/search/was/finding /qps/rest/3.0/count/was/finding /qps/rest/3.0/ignore/was/finding /qps/rest/3.0/activate/was/finding /qps/rest/3.0/editSeverity/was/finding /qps/rest/3.0/restoreSeverity/was/finding /qps/rest/3.0/retest/was/finding
New or Updated APIs	Updated
DTD or XSD changes	Yes

We have now introduced 36-bit unique ID (uniqueId) for each finding. The ID would be unique for every finding. Earlier, the combination of three fields namely: finding ID, finding type and finding category would make a finding unique. Now, with the implementation of uniqueId, you can easily distinguish every finding.

Input Parameters

We have introduced a new parameter for universal unique ID.

Parameter	Description
uniqueId={value}	The 36-bit unique id assigned to the finding. For example: <Finding> <id>132990</id> <uniqueId>8a2c4d51-6d28-2b92-e053- 2943720a74ab</uniqueId> <qid>150004</qid> ...

Sample - Get details of finding

If you search for a finding using unique ID (uniqueId), the count will always be one.

API request:

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

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

Request POST Data (file.xml):

```
<ServiceRequest>
  <filters>
    <Criteria field="uniqueId" operator="EQUALS">8a2c4d51-6d28-2b92-e053-
2943720a74ab</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>
</ServiceResponse>
```

Sample - Get details of finding using uniqueId

You can fetch details of a finding using uniqueId.

API request:

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/finding/8a2c4d51-6d28-
2b92-e053-2943720a74ab"
```

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>132990</id>
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
      <qid>150004</qid>
      <name>
        <![CDATA[Path-Based Vulnerability]]>
      </name>
      <type>VULNERABILITY</type>
      <findingType>QUALYS</findingType>
      <group>PATH</group>
      <cwe>
        <count>1</count>
      </cwe>
    </Finding>
  </data>
</ServiceResponse>
```

```
        <long>22</long>
      </list>
    </cwe>
    ...
    <isIgnored>>false</isIgnored>
    <retest/>
  </Finding>
</data>
</ServiceResponse>
```

Sample - Search finding using uniqueId

As every uniqueId is unique, using uniqueId, you could search for the exact 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>
  <filters>
    <Criteria field="uniqueId" operator="EQUALS">8a2c4d51-6d28-2b92-e053-2943720a74ab</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/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>>false</hasMoreRecords>
  <data>
    <Finding>
      <id>132990</id>
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
      <qid>150004</qid>
      <name>
        <![CDATA[Path-Based Vulnerability]]>
      </name>
      <type>VULNERABILITY</type>
      <findingType>QUALYS</findingType>
```

```
        <cwe>
          <count>1</count>
          <list>
            <long>22</long>
          </list>
        </cwe>
        ...
      </webApp>
      <isIgnored>>false</isIgnored>
      <retest/>
    </Finding>
  </data>
</ServiceResponse>
```

Sample - Ignore finding using uniqueId

As every uniqueId is unique, using uniqueId, you could ignore the exact finding.

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-  
binary @-  
"https://qualysapi.qualys.com/qps/rest/3.0/ignore/was/finding/8a2c4d51-  
6d28-2b92-e053-2943720a74ab" < file.xml
```

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

Request POST Data (file.xml):

```
<ServiceRequest>
  <data>
    <Finding>
      <ignoredReason>FALSE_POSITIVE</ignoredReason>
      <ignoredComment>test</ignoredComment>
    </Finding>
  </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/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Finding>
      <id>132990</id>
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
```

```
</Finding>  
</data>  
</ServiceResponse>
```

Sample - Activate a finding using uniqueId

As every uniqueId is unique, using uniqueId, you could ignore the exact finding.

API request:

```
curl -n -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/3.0/activate/was/finding/8a2c4d51-  
6d28-2b92-e053-2943720a74ab"
```

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>132990</id>  
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>  
    </Finding>  
  </data>  
</ServiceResponse>
```

Sample - Edit severity of a finding using uniqueId

As every uniqueId is unique, using uniqueId, you could edit the severity 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/editSeverity/was/finding/" <  
file.xml
```

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

Request POST Data (file.xml):

```
<ServiceRequest>  
  <data>  
    <Finding>  
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>  
      <severityComment>Test comment API</severityComment>
```

```
        <severity>3</severity>
      </Finding>
    </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/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Finding>
      <id>132990</id>
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
    </Finding>
  </data>
</ServiceResponse>
```

Sample - Restore severity of a finding using uniqueId

As every uniqueId is unique, you could restore the severity of specific finding.

API request:

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

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

Request POST Data (file.xml):

```
<ServiceRequest>
  <data>
    <Finding>
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
      <severityComment>Restoring default severity</severityComment>
      <severity>3</severity>
    </Finding>
  </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/was/finding.xsd">  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <Finding>  
      <id>132990</id>  
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>  
    </Finding>  
  </data>  
</ServiceResponse>
```

Sample - Retest a finding using uniqueId

API request:

```
curl -n -u "USERNAME:PASSWORD"  
"https://qualysapi.qualys.com/qps/rest/3.0/retest/was/finding/8a2c4d51-6d28-2b92-e053-2943720a74ab"
```

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/portal-api/xsd/3.0/was/finding.xsd"  
  <responseCode>SUCCESS</responseCode>  
  <count>1</count>  
  <data>  
    <Finding><id>1728792</id></Finding>  
    <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>  
  </data>  
</ServiceResponse>
```

XSD Update

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">

  <!-- REQUEST -->
  <xs:element name="ServiceRequest">
  ...
<xs:complexType name="Finding">
  <xs:all>
  <xs:element name="id" type="xs:long" minOccurs="0"/>
    <xs:element name="uniqueId" type="uniqueId" minOccurs="0"/>
  <xs:element name="qid" type="xs:long" minOccurs="0"/>
  <xs:element name="name" type="Cdata" minOccurs="0"/>
  <xs:element name="type" minOccurs="0">
  ...
<xs:complexType name="Cdata">
  <xs:simpleContent>
    <xs:extension base="xs:string"/>
  </xs:simpleContent>
</xs:complexType>
  <xs:simpleType name="uniqueId">
    <xs:restriction base="xs:string">
      <xs:pattern
        value="[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-
9]{4}-[a-f0-9]{12}" />
      </xs:restriction>
    </xs:simpleType>
  <xs:complexType name="UrlList">
  <xs:all>
  <xs:element name="count" type="xs:int" minOccurs="0"/>
  <xs:element name="list" minOccurs="0" >
    <xs:complexType>
    <xs:sequence>
    <xs:element name="Url" type="Url" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    </xs:complexType>
  </xs:element>
  </xs:all>
</xs:complexType>
  ...
  </xs:complexType>
</xs:schema>

```