Qualys Cloud Platform v2.x

API Release Notes

Version 2.41

September 5, 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: New DNS Override Support

HTTP Profile API: Support for Keeping/Removing Accept Encoding Header Field in Request Header

New API to Search for Security Events

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 Canada Platform 1	https://qualysapi.qg1.apps.qualys.ca
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.

WAS API: New DNS Override Support

API affected	/qps/rest/3.0/get/was/dnsoverride/{id} [GET] /qps/rest/3.0/count/was/dnsoverride/ [GET/POST} /qps/rest/3.0/search/was/dnsoverride/ [POST] /qps/rest/3.0/create/was/dnsoverride/ [POST] /qps/rest/3.0/update/was/dnsoverride/{id} [POST] /qps/rest/3.0/delete/was/dnsoverride/{id} [POST]
New or Updated APIs	New
DTD or XSD changes	Yes

By default we'll use the DNS for the web application URL to crawl the web app and perform scanning. If you provide a DNS override record through our new API, we'll use the mappings in your record instead.

There a few reasons you might want to do this. For example your web application does not have a DNS entry since it's in a non-production environment. Or the web application may have a different IP address in a non-production environment (e.g. development or QA) than in production.

Input Parameters

Parameter	Description

DnsMapping (keyword)	Use to configure the DNS override setting through API. You need to specify the hostname or FQDN and the corresponding IP address to be preferred for scanning.
	Example:
	`<set>`
	`<DnsMapping>`
	`<hostName>test</hostName>`
	`<ipAddress>2.3.4.5</ipAddress>`
	`</DnsMapping>`
	`</set>`

	When you create a new DNS override, ensure: -Name (Required): Name should be unique. -Tags: The tag id should be valid and in scope of current user. Use only `<Set>` tag. -Mappings (Required): Each mapping must have hostName and IpAddress in valid format. Use only <Set> tag. -Comments: Only `<Set>` with 1 comment is allowed with maximum length 2048 characters.
	When you update an DNS override, ensure: -Name: In case of name update, the updated name should be unique. -Id is required. -At lease one of the following should be present other than id: Name, owner, tags, comments, mappings -Tags: The `<set>` and `<Add>`/ `<Removed>` tags are mutually exclusive. Either use `<set>` or `<Add>` and `<Removed>`. - Mappings: The `<set>` and `<Add>`/ `<Removed>` tags are mutually exclusive. Either use `<set>` or `<Add>` and `<Removed>`.

Sample 1 - Get details of the DNS Override

Let us fetch details of DNS override. Ensure that you do not add any data or filter in the request.

API request:

```
curl -u "USERNAME:PASSWORD" " -X GET -H "Content-type: text/xml"
"https://qualysapi.qualys.com//qps/rest/3.0/get/was/dnsoverride/57020"
```

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/dnsoverride.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <DnsOverride>
            <id>57020</id>
            <name>
                <![CDATA[Test API DNS Record4]]>
            </name>
            ...
            <mappings>
                <count>3</count>
                <list>
                    <DnsMapping>
                        <hostName>host_1</hostName>
                        <ipAddress>1.2.3.7</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_2</hostName>
                        <ipAddress>1.2.3.5</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_3</hostName>
                        <ipAddress>1.2.3.5</ipAddress>
                    </DnsMapping>
                </list>
            </mappings>
        </DnsOverride>
    </data>
</ServiceResponse>
```

Sample 2 - Get the count for DNS Override

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/dnsoverride/" <
file.xml
Note: "file.xml" contains the request POST data.
```

Request POST Data (file.xml):

```
<ServiceRequest>
     <filters>
           <Criteria field="name" operator="CONTAINS">Test API</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/dnsoverride.xsd">
<responseCode>SUCCESS</responseCode>
    <count>6</count>
</ServiceResponse>
```

Sample 3 - Search for DNS Override

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"--data-
binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/dnsoverride/" <
file.xml
Note: "file.xml" contains the request POST data.
```

Request POST Data (file.xml):

```
<ServiceRequest>
     <filters>
           <Criteria field="name" operator="CONTAINS">Test API</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/xsd/3.0/was/dn
soverride.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>6</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <DnsOverride>
            <id>56420</id>
            <name>
                <![CDATA[Test API DNS Record]]>
            </name>
            <owner>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <tags>
                <count>0</count>
            </tags>
            <createdDate>2019-08-12T13:33:04Z</createdDate>
```

```
            <updatedDate>2019-08-12T13:33:04Z</updatedDate>
        </DnsOverride>
        <DnsOverride>
            <id>56422</id>
            <name>
                <![CDATA[Test API Dns Record1]]>
            </name>
            <owner>
                <id>1056860</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <tags>
                <count>0</count>
            </tags>
            <createdDate>2019-08-12T13:58:59Z</createdDate>
            <updatedDate>2019-08-12T13:58:59Z</updatedDate>
        </DnsOverride>
        ....
    </data>
</ServiceResponse>
```

Sample 4 - Create DNS Override

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/dnsoverride/" <
file.xml
Note: "file.xml" contains the request POST data.
```

Request POST Data (file.xml):

```
<ServiceRequest>
    <data>
        <DnsOverride>
            <name><![CDATA[DNS Record]]></name>
            <mappings>
                <set>
                    <DnsMapping>
                        <hostName>host_1</hostName>
                        <ipAddress>2.3.4.5</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_2</hostName>
                        <ipAddress>1.2.3.4</ipAddress>
                    </DnsMapping>
```

```
                </set>
            </mappings>
            <tags>
                <set>
                    <Tag>
                        <id>8993614</id>
                    </Tag>
                    <Tag>
                        <id>8876615</id>
                    </Tag>
                </set>
            </tags>
        </DnsOverride>
    </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/dnsoverride.xsd">
     <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <DnsOverride>
            <id>57220</id>
            <name>
                <![CDATA[DNS Record]]>
            </name>
            ...
            <mappings>
                <count>2</count>
                <list>
                    <DnsMapping>
                        <hostName>host_1</hostName>
                        <ipAddress>2.3.4.5</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_2</hostName>
                        <ipAddress>1.2.3.4</ipAddress>
                    </DnsMapping>
                </list>
            </mappings>
        </DnsOverride>
    </data>
</ServiceResponse>
```

Sample 5 - Update DNS Override (using set tag)

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/dnsoverride/" <
file.xml
Note: "file.xml" contains the request POST data.
```

Request POST Data (file.xml):

```
<ServiceRequest>
    <data>
        <DnsOverride>
            <name><![CDATA[DNS Record]]></name>
            <mappings>
                <set>
                    <DnsMapping>
                        <hostName>host_1</hostName>
                        <ipAddress>2.3.4.5</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_2</hostName>
                        <ipAddress>1.2.3.4</ipAddress>
                    </DnsMapping>
                </set>
            </mappings>
            <tags>
                <set>
                    <Tag>
                        <id>8993614</id>
                    </Tag>
                    <Tag>
                        <id>8876615</id>
                    </Tag>
                </set>
            </tags>
         </DnsOverride>
    </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/dnsoverride.xsd">
   <responseCode>SUCCESS</responseCode>
    <count>1</count>
```

```
    <data>
        <DnsOverride>
            <id>57020</id>
        </DnsOverride>
    </data>
</ServiceResponse>
```

Sample 6 - Update DNS Override (using add and remove tag)

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/dnsoverride/" <
file.xml
Note: "file.xml" contains the request POST data.
```

Request POST Data (file.xml):

```
<ServiceRequest>
    <data>
        <DnsOverride>
            <name><![CDATA[DNS Record]]></name>
            <mappings>
                <remove>
                    <DnsMapping>
                        <hostName>host_1</hostName>
                        <ipAddress>1.2.3.4</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_2</hostName>
                        <ipAddress>1.2.3.6</ipAddress>
                    </DnsMapping>
                </remove>
                <add>
                    <DnsMapping>
                        <hostName>host_3</hostName>
                        <ipAddress>1.2.3.5</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_4</hostName>
                        <ipAddress>1.2.3.7</ipAddress>
                    </DnsMapping>
                </add>
            </mappings>
            <tags>
                <set>
                    <Tag>
                        <id>8993614</id>
                    </Tag>
```

```
                    <Tag>
                        <id>8876615</id>
                    </Tag>
                </set>
            </tags>
        </DnsOverride>
    </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/dnsoverride.xsd">
   <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <DnsOverride>
            <id>57020</id>
        </DnsOverride>
    </data>
</ServiceResponse>
```

Sample 7 - Delete DNS Override

API request:

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"--data-
binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/dnsoverride/" <
file.xml
Note: "file.xml" contains the request POST data.
```

Request POST Data (file.xml):

```
<ServiceRequest>
    <filters>
        <Criteria field="id" operator="EQUALS">57020</Criteria>
    </filters>
    <data>
        <DnsOverride>
            <id>57220</id>
        </DnsOverride>
    </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/dnsoverride.xsd">
   <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <DnsOverride>
            <id>57220</id>
        </DnsOverride>
    </data>
</ServiceResponse>
```

New XSD (dnsoverride.xsd)

The new XSD is available at:
https://qualysapi.qualys.com/qps/xsd/3.0/was/dnsoverride.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>
            <xs:all>
                <xs:element name="filters" type="ServiceRequestFilters"
minOccurs="0"/>
                <xs:element name="preferences"
type="ServiceRequestPreferences" minOccurs="0"/>
                <xs:element name="data" type="ServiceRequestData"
minOccurs="0"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
     <xs:complexType name="ServiceRequestFilters">
        <xs:sequence>
            <xs:element name="Criteria" type="Criteria"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
     <xs:complexType name="Criteria">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="field" type="xs:string"/>
                <xs:attribute name="operator">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="CONTAINS"/>
                            <xs:enumeration value="IN"/>
                            <xs:enumeration value="EQUALS"/>
                            <xs:enumeration value="NOT EQUALS"/>
                            <xs:enumeration value="GREATER"/>
                            <xs:enumeration value="LESSER"/>
                            <xs:enumeration value="NONE"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
     <xs:complexType name="ServiceRequestPreferences">
        <xs:all>
```

```
            <xs:element name="startFromId" type="xs:long" minOccurs="0"/>
            <xs:element name="startFromOffset" type="xs:int"
minOccurs="0"/>
            <xs:element name="limitResults" type="xs:int" minOccurs="0"/>
        </xs:all>
    </xs:complexType>
     <xs:complexType name="ServiceRequestData">
        <xs:sequence>
            <xs:element name="DnsOverride" type="DnsOverride"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
     <!-- RESPONSE -->
    <xs:element name="ServiceResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="responseCode" type="ResponseCode"/>
                <xs:element name="responseErrorDetails"
type="ResponseErrorObject" minOccurs="0"/>
                <xs:element name="count" type="xs:int" minOccurs="0"/>
                <xs:element name="hasMoreRecords" type="xs:boolean"
minOccurs="0"/>
                <xs:element name="lastId" type="xs:long" minOccurs="0"/>
                <xs:element name="data" type="ServiceResponseData"
minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
     <xs:simpleType name="ResponseCode">
        <xs:restriction base="xs:string">
            <xs:enumeration value="AUTH_CREDENTIALS_NEEDED"/>
            <xs:enumeration value="CANNOT_BE_NULL"/>
            <xs:enumeration value="INVALID_XML"/>
            <xs:enumeration value="INVALID_CREDENTIALS"/>
            <xs:enumeration value="INVALID_API_VERSION"/>
            <xs:enumeration value="INVALID_PARAM"/>
            <xs:enumeration value="INVALID_URL"/>
            <xs:enumeration value="INVALID_REQUEST"/>
            <xs:enumeration value="NOT_FOUND"/>
            <xs:enumeration value="OTHER_ERROR"/>
            <xs:enumeration value="OPERATION_NOT_SUPPORTED"/>
            <xs:enumeration value="EVALUATION_EXPIRED"/>
            <xs:enumeration value="JMS_SERVER_DOWN"/>
            <xs:enumeration value="RMI_SERVER_DOWN"/>
            <xs:enumeration value="SUCCESS"/>
            <xs:enumeration value="STILL_PROCESSING"/>
            <xs:enumeration value="UNAUTHORIZED"/>
            <xs:enumeration value="UNAUTHORIZED_DESTINATION_APPS"/>
            <xs:enumeration value="UNIDENTIFIED_PRODUCER"/>
```

```
            <xs:enumeration value="UNKNOWN_OBJECT"/>
        </xs:restriction>
    </xs:simpleType>
     <xs:complexType name="ResponseErrorObject">
        <xs:sequence>
            <xs:element name="errorMessage" type="xs:string"/>
            <xs:element name="errorResolution" type="xs:string"
minOccurs="0"/>
            <xs:element name="internalErrorCodeId" type="xs:int"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
     <xs:complexType name="ServiceResponseData">
        <xs:sequence>
            <xs:element name="DnsOverride" type="DnsOverride"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <!-- DATA -->
     <xs:complexType name="DnsOverride">
        <xs:all>
            <xs:element name="id" type="xs:long" minOccurs="0"/>
            <xs:element name="name" type="Cdata" minOccurs="0"/>
            <xs:element name="owner" type="User" minOccurs="0"/>
            <xs:element name="tags" type="TagList" minOccurs="0"/>
           <xs:element name="comments" type="CommentList" minOccurs="0"/>
            <xs:element name="createdDate" type="xs:dateTime"
minOccurs="0"/>
            <xs:element name="createdBy" type="User" minOccurs="0"/>
            <xs:element name="updatedDate" type="xs:dateTime"
minOccurs="0"/>
            <xs:element name="updatedBy" type="User" minOccurs="0"/>
           <xs:element name="mappings" type="DnsMappings" minOccurs="0"/>
        </xs:all>
    </xs:complexType>
     <xs:complexType name="Cdata">
        <xs:simpleContent>
            <xs:extension base="xs:string"/>
        </xs:simpleContent>
    </xs:complexType>
     <xs:complexType name="User">
        <xs:all>
            <xs:element name="id" type="xs:long"/>
            <xs:element name="username" type="xs:string" minOccurs="0"/>
            <xs:element name="firstName" type="Cdata" minOccurs="0"/>
            <xs:element name="lastName" type="Cdata" minOccurs="0"/>
        </xs:all>
    </xs:complexType>
     <xs:complexType name="TagList">
```

```
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Tag" type="Tag" minOccurs="0"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="set" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Tag" type="Tag"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="add" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Tag" type="Tag"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="remove" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Tag" type="Tag"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="update" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Tag" type="Tag"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>
     <xs:complexType name="Tag">
        <xs:all>
            <xs:element name="id" type="xs:long"/>
            <xs:element name="name" type="Cdata" minOccurs="0"/>
        </xs:all>
```

```
    </xs:complexType>
     <xs:complexType name="Comment">
        <xs:all>
            <xs:element name="contents" type="Cdata"/>
            <xs:element name="author" type="User" minOccurs="0"/>
            <xs:element name="createdDate" type="xs:dateTime"
minOccurs="0"/>
        </xs:all>
    </xs:complexType>
     <xs:complexType name="CommentList">
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Comment" type="Comment"
minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="set" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Comment" type="Comment"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="add" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Comment" type="Comment"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>
     <xs:complexType name="DnsMappings">
        <xs:all>
            <xs:element name="count" type="xs:int" minOccurs="0"/>
            <xs:element name="list" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="DnsMapping" type="DnsMapping"
minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
```

```
            <xs:element name="set" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="DnsMapping" type="DnsMapping"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="add" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="DnsMapping" type="DnsMapping"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="remove" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="DnsMapping" type="DnsMapping"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>
     <xs:complexType name="DnsMapping">
        <xs:sequence>
            <xs:element name="hostName" type="xs:string" minOccurs="0"
maxOccurs="1" />
            <xs:element name="ipAddress" type="xs:string" maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
 </xs:schema>
```

HTTP Profile API: Support for Keeping/Removing Accept Encoding Header Field in Request Header

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

You can now create an HTTP profile for your web application to specify the WAF application to either retain or remove the Accept Encoding header field in requests. If you choose to retain the Accept Encoding header field then WAF will keep the header field in requests that contain this header field while forwarding the requests to your web application. By default, WAF will remove this header field.

Input Parameters

New parameter is described below.

Parameter	Description
keepAcceptEncoding (Boolean)	If this is set to true, the Accept Encoding header field will be kept in the request. By default, this parameter is set to false.

Sample 1 - Get HTTP Profile to view accept encoding header field value

API request:

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/get/waf/httpprofile/4401"
```

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/2.0/wa
f/httpprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <HTTPProfile>
            <id>4401</id>
            <uuid>4ffd8f5e-529d-4a38-9cac-7fb805962a18</uuid>
            <name>
                <![CDATA[my profile from API]]>
            </name>
```

```
                <description>
                    <![CDATA[desc of proto profile]]>
                </description>
                <owner>
                    <id>3988443</id>
                    <username>john_doe</username>
                    <firstname>john</firstname>
                    <lastname>doe</lastname>
                </owner>
                <created>2017-04-26T15:25:26Z</created>
                <createdBy>
                    <id>3988443</id>
                    <username>john_doe<</username>
                    <firstname>john</firstname>
                    <lastname>doe</lastname>
                </createdBy>
                <updated>2017-04-26T15:25:26Z</updated>
                <updatedBy>
                    <id>3988443</id>
                    <username>john_doe</username>
                    <firstname>john</firstname>
                    <lastname>doe</lastname>
                </updatedBy>
                <system>false</system>
                <requestMethod/>
                <keepAcceptEncoding>false</keepAcceptEncoding>
                <requestHeader>
...
```

Sample 2 - Create HTTP Profile to retain accept encoding header field in requests

Let us create a customized HTTP profile and set the "keep accept encoding" header field to true.

API request:

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

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

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <HTTPProfile>
            <name>my profile</name>
            <description>desc of proto profile</description>
```

```
                <requestMethod>
                    <allowAll>
                        <detectInvalid>false</detectInvalid>
                        <detectTraceTrack>false</detectTraceTrack>
                    </allowAll>
                </requestMethod>
                <keepAcceptEncoding>true</keepAcceptEncoding>
                <detectProtocolAnomalies>false</detectProtocolAnomalies>
                <requestHeader>
                    <detectInvalid>false</detectInvalid>
                    <detectRepeated>false</detectRepeated>
                    <detectChunked>false</detectChunked>
                </requestHeader>
        ...
```

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/httpprofile.xsd">
...
        <HTTPProfile>
            <id>4801</id>
            <uuid>d6f60c16-5146-477d-a005-a2d182cb0632</uuid>
            <name>
                <![CDATA[my profile from API 11]]>
            </name>
            <description>
                <![CDATA[desc of proto profile]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>john</firstname>
                <lastname>doe</lastname>
            </owner>
            <created>2017-05-04T10:04:54Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>john</firstname>
                <lastname>doe</lastname>
            </createdBy>
            <updated>2017-05-04T10:04:54Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>john</firstname>
                <lastname>doe</lastname>
```

```
                </updatedBy>
                <system>false</system>
                <requestMethod>
                    <allowAll>
                        <detectInvalid>false</detectInvalid>
                        <detectTraceTrack>false</detectTraceTrack>
                    </allowAll>
                </requestMethod>
                <keepAcceptEncoding>true</keepAcceptEncoding>
                <requestHeader>
                    <detectInvalid>false</detectInvalid>
                    <detectRepeated>false</detectRepeated>
                    <detectChunked>false</detectChunked>
                </requestHeader>
...
```

Sample 3 - Update HTTP Profile to remove the accept encoding header field in requests

Let us update an HTTP profile to remove the "keep accept encoding" header field in the requests by setting its value to false.

API request:

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

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

Request POST Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <HTTPProfile>
            <name>my profile</name>
            <description>desc of proto profile</description>
            <requestMethod>
                <allowAll>
                    <detectInvalid>false</detectInvalid>
                    <detectTraceTrack>false</detectTraceTrack>
                </allowAll>
            </requestMethod>
            <keepAcceptEncoding>false</keepAcceptEncoding>
            <detectProtocolAnomalies>false</detectProtocolAnomalies>
            <requestHeader>
                <detectInvalid>false</detectInvalid>
                <detectRepeated>false</detectRepeated>
          ...
```

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/httpprofile.xsd">
...
        <HTTPProfile>
            <id>4801</id>
            <uuid>d6f60c16-5146-477d-a005-a2d182cb0632</uuid>
            <name>
                <![CDATA[my profile from API 11]]>
            </name>
            <description>
                <![CDATA[desc of proto profile]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>john</firstname>
                <lastname>doe</lastname>
            </owner>
            <created>2017-05-04T10:04:54Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>john</firstname>
                <lastname>doe</lastname>
            </createdBy>
            <updated>2017-05-04T10:04:54Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>john</firstname>
                <lastname>doe</lastname>
            </updatedBy>
            <system>false</system>
            <requestMethod>
                <allowAll>
                    <detectInvalid>false</detectInvalid>
                    <detectTraceTrack>false</detectTraceTrack>
                </allowAll>
            </requestMethod>
            <keepAcceptEncoding>false</keepAcceptEncoding>
            <requestHeader>
                <detectInvalid>false</detectInvalid>
                <detectRepeated>false</detectRepeated>
                <detectChunked>false</detectChunked>
            ...
```

HTTP Profile XSD update

(qps/xsd/2.0/waf/httpprofile.xsd)

A new element in bold "keepAcceptEncoding" added to the XSD.

```
...
        <xs:complexType name="HTTPProfile">
        <xs:sequence>
            <xs:element name="id" type="xs:long" minOccurs="0" />
            <xs:element name="uuid" type="Uuid" minOccurs="0" />
            <xs:element name="name" type="Cdata" minOccurs="0" />
            <xs:element name="description" type="Cdata" minOccurs="0" />
            <xs:element name="owner" type="User" minOccurs="0" />
           <xs:element name="created" type="xs:dateTime" minOccurs="0" />
            <xs:element name="createdBy" type="User" minOccurs="0" />
           <xs:element name="updated" type="xs:dateTime" minOccurs="0" />
            <xs:element name="updatedBy" type="User" minOccurs="0" />
            <xs:element name="system" type="xs:boolean" minOccurs="0" />
            <xs:element name="requestMethod" minOccurs="0">
                <xs:complexType>
                    <xs:choice>
                        <xs:sequence>
                            <xs:element name="allowAll">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="detectInvalid"
type="xs:boolean" minOccurs="0" />
                                          <xs:element name="detectTraceTrack"
type="xs:boolean" minOccurs="0" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                        <xs:sequence>
                            <xs:element name="denyAll" type="Cdata" />
                        </xs:sequence>
                    </xs:choice>
                </xs:complexType>
            </xs:element>
            <xs:element name="keepAcceptEncoding" type="xs:boolean"
minOccurs="0" maxOccurs="1" />
            <xs:element name="requestHeader" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                       <xs:element name="detectInvalid" type="xs:boolean"
minOccurs="0" />
                      <xs:element name="detectRepeated" type="xs:boolean"
minOccurs="0" />
                       <xs:element name="detectChunked" type="xs:boolean"
```

```
minOccurs="0" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
    ...
```

New API to Search for Security Events

API affected	/qps/rest/2.0/search/waf/eventlog
New or Updated APIs	New
DTD or XSD changes	New

You can now search for security events detected for your web application in the event log using the search filters provided by the Eventlog API.

Permission Required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and web apps licensed for WAF and within the user's scope.

Input Parameters

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. If no parameters are passed then this API will return only the current day events. All dates must be entered in UTC date/time format.

Parameter	Description
transactionId (UUID)	Transaction identifier of the event.
action (Text)	Action taken for a event. Valid values are: NOT_BLOCKED, REQUEST_BLOCKED, RESPONSE_BLOCKED
blocked (Boolean)	If true, this parameter is evaluated as "action" parameter with value as "REQUEST_BLOCKED". If false, this parameter is evaluated as "action" parameter with value "REQUEST_ALLOWED".
flagged (Boolean)	True if the event is flagged.
archived (Boolean)	True if the event is archived.
falsePositive (Boolean)	True if the event is marked as false positive.
notApplicable (Boolean)	True if the event is marked not applicable.
webApplication (Text)	Web application for which event is generated.
source.ip (Text)	IP of the event sources.
source.country (Text)	Country from where events originated.
occurred (Date)	Date and time when events occurred.
responseCode (Integer)	Response code generated for events.
qids (Integer)	QIDs of the detections for the events.
threatLevel (Text)	Threat level of the detections for the events. Valid values are: LOW, MEDIUM, HIGH.

Supported Operators

For searching events, operators supported by the elements are:

- transactionId, action, blocked, source.ip, source.country, responseCode, qids, threatLevel support only EQUALS operator.

- flagged, archived, falsePositive, notApplicable support EQUALS and NOT EQUALS operators.

- occurred supports EQUALS, GREATER and LESSER operators.

- qids supports only IN operator.

- webApplication supports only CONTAINS operator.

Sample 1 - Search for event logs using the search parameters

API request:

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

Request POST data :

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <preferences>
        <startFromOffset>1</startFromOffset>
        <limitResults>1</limitResults>
    </preferences>
  <filters>
    <!-- Criteria field="transactionId" operator="EQUALS">6b1b0a90-e9ab-
411e-8b53-453ee5239a81</Criteria -->
    <Criteria field="action"
operator="EQUALS">REQUEST_BLOCKED</Criteria><!-- NOT_BLOCKED,
REQUEST_BLOCKED, RESPONSE_BLOCKED -->
    <Criteria field="flagged" operator="EQUALS">false</Criteria>
    <Criteria field="archived" operator="EQUALS">false</Criteria>
    <Criteria field="falsePositive" operator="EQUALS">false</Criteria>
    <Criteria field="notApplicable" operator="EQUALS">false</Criteria>
    <Criteria field="webApplication" operator="CONTAINS">Docker</Criteria>
    <!--Criteria field="source.ip"
operator="EQUALS">204.177.170.98</Criteria-->
    <Criteria field="source.country" operator="EQUALS">US</Criteria>
    <Criteria field="responseCode" operator="EQUALS">403</Criteria>
    <Criteria field="occurred" operator="LESSER">2019-07-
22T08:16:00Z</Criteria>
    <Criteria field="occurred" operator="GREATER">2019-07-
22T08:15:00Z</Criteria>
```

```
    <!-- Criteria field="qids" operator="IN">226018</Criteria -->
  </filters>
</ServiceRequest>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://localhost:38080/portal-
api/xsd/2.0/waf/eventlog.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>true</hasMoreRecords>
    <data>
        <EventLog>
            <uuid>50f1fc8b-3fc1-4cfb-b238-78012307450c</uuid>
            <location>
                <![CDATA[/path1?ldap=test)]]>
            </location>
            <archived>false</archived>
            <blocked>true</blocked>
            <falsePositive>false</falsePositive>
            <flagged>false</flagged>
            <notApplicable>false</notApplicable>
            <occurred>2019-07-22T08:15:09Z</occurred>
            <source>
                <country>US</country>
                <ip>204.177.170.98</ip>
                <latitude>37.751</latitude>
                <longitude>-97.822</longitude>
                <port>60664</port>
            </source>
            <eventDetails>
                <EventDetail>
                    <description>LDAPi: Common fingerprinting test
detected.</description>
                    <qid>226003</qid>
                    <type>ALERT</type>
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>25</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>30</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>SQLi: Burp SQL keywords fuzzing
detected.</description>
                    <qid>150003</qid>
                    <type>ALERT</type>
```

```
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>20</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>20</threatScore>
                </EventDetail>
                <EventDetail>
                   <description>XSS: Javascript function associated with
event or script.</description>
                    <qid>150001</qid>
                    <type>ALERT</type>
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>10</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>10</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>XSS: Heuristic cross-site scripting
detected - param1.</description>
                    <qid>150001</qid>
                    <type>ALERT</type>
                    <confidenceLevel>HIGH</confidenceLevel>
                    <confidenceScore>80</confidenceScore>
                    <threatLevel>HIGH</threatLevel>
                    <threatScore>70</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>XSS: Heuristic cross-site scripting
detected - param2.</description>
                    <qid>150001</qid>
                    <type>ALERT</type>
                    <confidenceLevel>HIGH</confidenceLevel>
                    <confidenceScore>80</confidenceScore>
                    <threatLevel>HIGH</threatLevel>
                    <threatScore>70</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>XSS: Script tag detected - &lt;script
alert(2)&lt;/script&gt;.</description>
                    <qid>150001</qid>
                    <type>ALERT</type>
                    <confidenceLevel>MEDIUM</confidenceLevel>
                    <confidenceScore>40</confidenceScore>
                    <threatLevel>MEDIUM</threatLevel>
                    <threatScore>40</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>XSS: Script tag detected - &lt;script
alert(2)&lt;/script&gt;.</description>
                    <qid>150001</qid>
```

```
                    <type>ALERT</type>
                    <confidenceLevel>MEDIUM</confidenceLevel>
                    <confidenceScore>40</confidenceScore>
                    <threatLevel>MEDIUM</threatLevel>
                    <threatScore>40</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>XSS: Javascript function detected -
alert(2).</description>
                    <qid>150001</qid>
                    <type>ALERT</type>
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>30</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>30</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>XSS: Javascript function detected -
alert(2).</description>
                    <qid>150001</qid>
                    <type>ALERT</type>
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>30</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>30</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>XSS: HTML tag detected - &lt;script
.</description>
                    <qid>150001</qid>
                    <type>ALERT</type>
                    <confidenceLevel>MEDIUM</confidenceLevel>
                    <confidenceScore>40</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>30</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>XSS: HTML tag detected - &lt;script
.</description>
                    <qid>150001</qid>
                    <type>ALERT</type>
                    <confidenceLevel>MEDIUM</confidenceLevel>
                    <confidenceScore>40</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>30</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>LFi: Unknown filesystem file detected -
.exe.</description>
```

```
                    <qid>150011</qid>
                    <type>ALERT</type>
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>30</confidenceScore>
                    <threatLevel>MEDIUM</threatLevel>
                    <threatScore>40</threatScore>
                </EventDetail>
                <EventDetail>
                   <description>RCE: Command detected at the beginning of
the string - cmd.exe.</description>
                    <qid>226008</qid>
                    <type>ALERT</type>
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>30</confidenceScore>
                    <threatLevel>MEDIUM</threatLevel>
                    <threatScore>60</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>XMLi: 2 XML tags detected.</description>
                    <qid>226018</qid>
                    <type>ALERT</type>
                    <confidenceLevel>MEDIUM</confidenceLevel>
                    <confidenceScore>40</confidenceScore>
                    <threatLevel>MEDIUM</threatLevel>
                    <threatScore>40</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>XMLi: 2 XML tags detected.</description>
                    <qid>226018</qid>
                    <type>ALERT</type>
                    <confidenceLevel>MEDIUM</confidenceLevel>
                    <confidenceScore>40</confidenceScore>
                    <threatLevel>MEDIUM</threatLevel>
                    <threatScore>40</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>Info: Transaction blocked by
engine.</description>
                    <qid>226016</qid>
                    <type>OBSERVATION</type>
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>0</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>0</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>Info: Threat level exceeded blocking
threshold (1).</description>
                    <qid>226016</qid>
```

```
                    <type>OBSERVATION</type>
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>0</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>0</threatScore>
                </EventDetail>
                <EventDetail>
                   <description>Info: Blocking transaction.</description>
                    <qid>226016</qid>
                    <type>OBSERVATION</type>
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>0</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>0</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>Info: Logging transaction.</description>
                    <type>ALERT</type>
                    <confidenceLevel>LOW</confidenceLevel>
                    <confidenceScore>0</confidenceScore>
                    <threatLevel>LOW</threatLevel>
                    <threatScore>0</threatScore>
                </EventDetail>
                <EventDetail>
                    <description>Detected: LDAPi, SQLi, XSS, LFi,
CommandExec, XMLi.</description>
                    <qid>226018</qid>
                    <type>ALERT</type>
                    <confidenceLevel>HIGH</confidenceLevel>
                    <confidenceScore>80</confidenceScore>
                    <threatLevel>HIGH</threatLevel>
                    <threatScore>70</threatScore>
                </EventDetail>
            </eventDetails>
            <responseCode>403</responseCode>
            <summary>XSS: Heuristic cross-site scripting detected -
param1.</summary>
            <threatLevel>HIGH</threatLevel>
            <threatScore>70</threatScore>
            <transactionDuration>16</transactionDuration>
            <transactionId>50F1FC8B-3FC1-4CFB-B238-
78012307450C</transactionId>
            <transactionDetails>
                <request>
                    <![CDATA[POST /path1?ldap=test) HTTP/1.1
Host: dip02.p29.eng.sjc01.qualys.com
X-Forwarded-For: 204.177.170.98
User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7
NSS/3.27.1 zlib/1.2.3 libidn/1.18 libssh2/1.4.2
```

```
Accept: */*
Content-Type: application/x-www-form-urlencoded
Cookie: _ga=cmd.exe; _mkto_trk=id:797-ENI-742&token:_mch-qualys.com-
1536168280266-16119
Content-Length: 65
X-Forwarded-For: 172.17.42.1


param1=<script>alert(2)</script>&param2=<script>alert(2)</script>]]>
                </request>
                <response>
                    <![CDATA[HTTP/1.0 403 Forbidden
Connection: close
Date: Mon, 22 Jul 2019 08:15:09 +0000 (UTC)
Content-Type: text/html


                    <!doctype html><html><head><meta charset="UTF-
8"><title>Access Denied</title></head><style type="text/css">.a{font-
family:"Lucida Grande","Lucida Sans Unicode","Lucida Sans","DejaVu
Sans",Verdana,sans-serif;font-size:15px;color:#3a3a3a;paddi...]]>
            </response>
        </transactionDetails>
        <webApps>
            <WebApp>
                <uuid>47df31b0-4846-4619-862c-74b018ae6ae9</uuid>
                <name>
                    <![CDATA[Docker web application]]>
                </name>
            </WebApp>
        </webApps>
    </EventLog>
</data>
</ServiceResponse>
```

New XSD (eventlog.xsd)

The new XSD is available at: <platform API server>/qps/xsd/2.0/waf/eventlog.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>
            <xs:all>
                <xs:element name="filters" type="ServiceRequestFilters"
minOccurs="0"/>
                <xs:element name="preferences"
type="ServiceRequestPreferences" minOccurs="0"/>
                <xs:element name="data" type="ServiceRequestData"
minOccurs="0"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="ServiceRequestFilters">
        <xs:sequence>
            <xs:element name="Criteria" type="Criteria"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="Criteria">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="field" type="xs:string"/>
                <xs:attribute name="operator">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="CONTAINS"/>
                            <xs:enumeration value="IN"/>
                            <xs:enumeration value="EQUALS"/>
                            <xs:enumeration value="NOT EQUALS"/>
                            <xs:enumeration value="GREATER"/>
                            <xs:enumeration value="LESSER"/>
                            <xs:enumeration value="NONE"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="ServiceRequestPreferences">
        <xs:all>
            <xs:element name="startFromId" type="xs:long" minOccurs="0"/>
```

```
            <xs:element name="startFromOffset" type="xs:int"
minOccurs="0"/>
            <xs:element name="limitResults" type="xs:int" minOccurs="0"/>
        </xs:all>
    </xs:complexType>
    <xs:complexType name="ServiceRequestData">
        <xs:sequence>
            <xs:element name="EventLog" type="EventLog" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <!-- RESPONSE -->
    <xs:element name="ServiceResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="responseCode" type="ResponseCode"/>
                <xs:element name="responseErrorDetails"
type="ResponseErrorObject" minOccurs="0"/>
                <xs:element name="count" type="xs:int" minOccurs="0"/>
                <xs:element name="hasMoreRecords" type="xs:boolean"
minOccurs="0"/>
                <xs:element name="lastId" type="xs:long" minOccurs="0"/>
                <xs:element name="data" type="ServiceResponseData"
minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:simpleType name="ResponseCode">
        <xs:restriction base="xs:string">
            <xs:enumeration value="AUTH_CREDENTIALS_NEEDED"/>
            <xs:enumeration value="CANNOT_BE_NULL"/>
            <xs:enumeration value="INVALID_XML"/>
            <xs:enumeration value="INVALID_CREDENTIALS"/>
            <xs:enumeration value="INVALID_API_VERSION"/>
            <xs:enumeration value="INVALID_PARAM"/>
            <xs:enumeration value="INVALID_URL"/>
            <xs:enumeration value="INVALID_REQUEST"/>
            <xs:enumeration value="NOT_FOUND"/>
            <xs:enumeration value="OTHER_ERROR"/>
            <xs:enumeration value="OPERATION_NOT_SUPPORTED"/>
            <xs:enumeration value="JMS_SERVER_DOWN"/>
            <xs:enumeration value="RMI_SERVER_DOWN"/>
            <xs:enumeration value="SUCCESS"/>
            <xs:enumeration value="STILL_PROCESSING"/>
            <xs:enumeration value="UNAUTHORIZED"/>
            <xs:enumeration value="UNAUTHORIZED_DESTINATION_APPS"/>
            <xs:enumeration value="UNIDENTIFIED_PRODUCER"/>
            <xs:enumeration value="UNKNOWN_OBJECT"/>
        </xs:restriction>
    </xs:simpleType>
```

```
    <xs:complexType name="ResponseErrorObject">
        <xs:sequence>
            <xs:element name="errorMessage" type="xs:string"/>
            <xs:element name="errorResolution" type="xs:string"
minOccurs="0"/>
            <xs:element name="internalErrorCodeId" type="xs:int"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceResponseData">
        <xs:sequence>
            <xs:element name="EventLog" type="EventLog" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <!-- DATA -->
    <xs:simpleType name="Uuid">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-
F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="Cdata">
        <xs:simpleContent>
            <xs:extension base="xs:string"/>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="User">
        <xs:sequence>
            <xs:element name="id" type="xs:long" minOccurs="1"
maxOccurs="1"/>
            <xs:element name="name" type="xs:string" minOccurs="1"
maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="Level">
        <xs:restriction base="xs:string">
            <xs:enumeration value="LOW"/>
            <xs:enumeration value="MEDIUM"/>
            <xs:enumeration value="HIGH"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="EventLog">
        <xs:sequence>
            <xs:element name="uuid" type="Uuid"/>
            <xs:element name="location" type="Cdata"/>
            <xs:element name="archived" type="xs:boolean" minOccurs="0"/>
```

```
            <xs:element name="blocked" type="xs:boolean" minOccurs="0"/>
            <xs:element name="falsePositive" type="xs:boolean"
minOccurs="0"/>
            <xs:element name="flagged" type="xs:boolean" minOccurs="0"/>
            <xs:element name="notApplicable" type="xs:boolean"
minOccurs="0"/>
           <xs:element name="occurred" type="xs:dateTime" minOccurs="0"/>
            <xs:element name="source">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="city" type="xs:string"
minOccurs="0"/>
                        <xs:element name="country" type="xs:string"
minOccurs="0"/>
                        <xs:element name="ip" type="xs:string"
minOccurs="0"/>
                        <xs:element name="latitude" type="xs:decimal"
minOccurs="0"/>
                        <xs:element name="longitude" type="xs:decimal"
minOccurs="0"/>
                        <xs:element name="port" type="xs:int"
minOccurs="0"/>
                        <xs:element name="region" type="xs:string"
minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="responseCode" type="xs:int" minOccurs="0"/>
            <xs:element name="qid" type="xs:long" minOccurs="0"/>
            <xs:element name="summary" type="xs:string" minOccurs="0"/>
            <xs:element name="threatLevel" type="Level" minOccurs="0"/>
            <xs:element name="threatScore" type="xs:int" minOccurs="0"/>
            <xs:element name="transactionDuration" type="xs:long"
minOccurs="0"/>
            <xs:element name="transactionId" type="Uuid" minOccurs="0"/>
            <xs:element name="policy" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                      <xs:element name="id" type="xs:long" minOccurs="0"/>
                       <xs:element name="uuid" type="Uuid" minOccurs="0"/>
                      <xs:element name="name" type="Cdata" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="webApps" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="WebApp" maxOccurs="unbounded">
                            <xs:complexType>
```

```
                                <xs:sequence>
                                    <xs:element name="id" type="xs:long"/>
                                     <xs:element name="uuid" type="Uuid"/>
                                    <xs:element name="name" type="Cdata"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="eventDetails" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="action" type="xs:string"
minOccurs="0"/>
                        <xs:element name="confidence" type="Level"
minOccurs="0"/>
                        <xs:element name="confidenceScore" type="xs:int"
minOccurs="0"/>
                        <xs:element name="description" type="xs:string"
minOccurs="0"/>
                        <xs:element name="qid" type="xs:long"
minOccurs="0"/>
                        <xs:element name="ruleId" type="xs:string"
minOccurs="0"/>
                        <xs:element name="severity" type="Level"
minOccurs="0"/>
                        <xs:element name="severityScore" type="xs:int"
minOccurs="0"/>
                        <xs:element name="suppression" type="xs:string"
minOccurs="0"/>
                        <xs:element name="tag" type="xs:string"
minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="type" type="xs:string"
minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="transactionDetails" minOccurs="0" >
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="request" type="Cdata"
minOccurs="0"/>
                        <xs:element name="response" type="Cdata"
minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
```

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

</xs:schema>
```