Web Application Firewall API

User Guide

Version 2.41 (2.12)

June 7, 2021

Table of Contents

Welcome

Web Application Firewall API

The Web Application Firewall (WAF) API supports integrating the Qualys Web Application Firewall solution into third party applications.

Modules supported

WAF

Authentication

Authentication to your Qualys account with valid Qualys credentials is required for making Qualys API requests to the Qualys API servers. Learn more about authentication to your Qualys account

Get API Notifications

We recommend you join our Community and subscribe to our API Notifications RSS Feeds for announcements and discussions.

https://community.qualys.com/community/developer/notifications-api

About Qualys

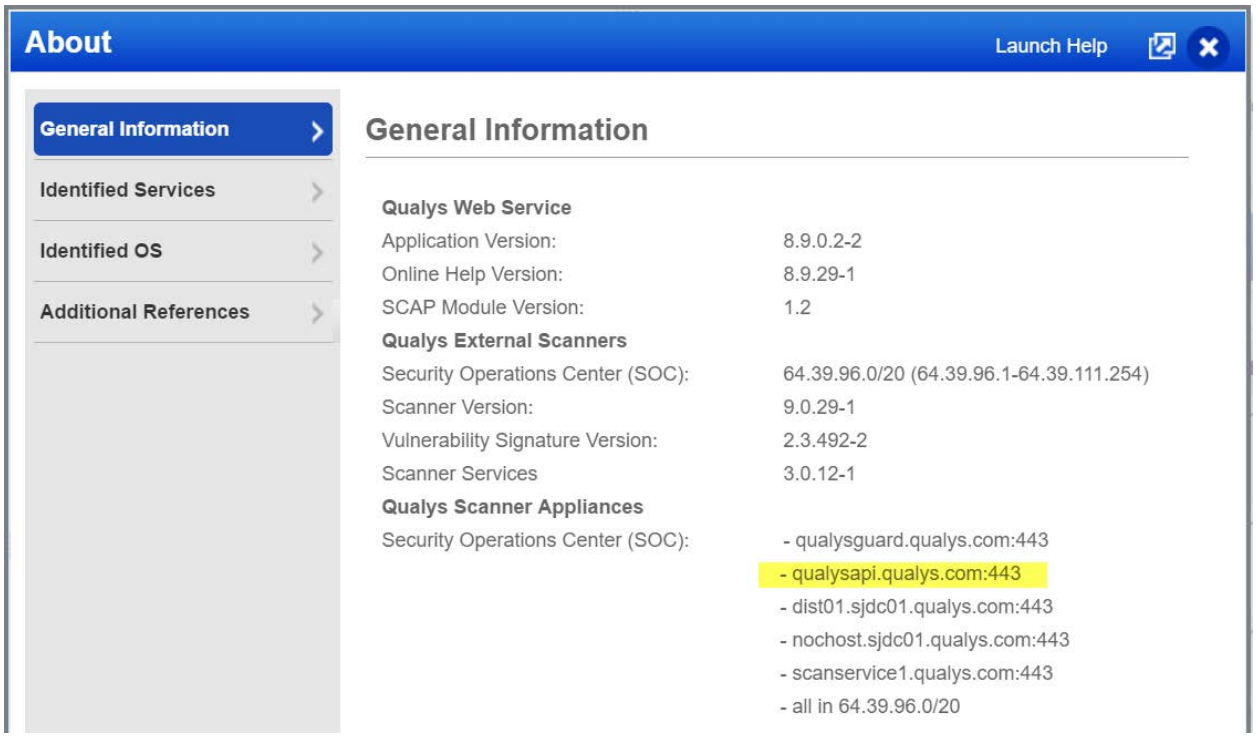
Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of cloud-based security and compliance solutions. The Qualys Cloud Platform and its integrated Cloud Apps deliver businesses critical security intelligence continuously, enabling them to automate the full spectrum of auditing, compliance and protection for IT systems and web applications on premises, on endpoints and elastic clouds. For more information, please visit www.qualys.com

URL to Qualys API server

Qualys maintains multiple Qualys Cloud Platforms. The API server URL that you should use for API requests depends on the platform where your Qualys 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	https://qualysapi.<customer_base_url>

Looking for your API server URL for your account? You can find this easily. Just log in to your Qualys account and go to Help > About. You'll see this information under Security Operations Center (SOC).


About
Launch Help
General Information
Identified Services
Identified OS
Additional References
General Information
Qualys Web Service
Application Version: 8.9.0.2-2
Online Help Version: 8.9.29-1
SCAP Module Version: 1.2
Qualys External Scanners
Security Operations Center (SOC): 64.39.96.0/20 (64.39.96.1-64.39.111.254)
Scanner Version: 9.0.29-1
Vulnerability Signature Version: 2.3.492-2
Scanner Services 3.0.12-1
Qualys Scanner Appliances
Security Operations Center (SOC): - qualysguard.qualys.com:443
- qualysapi.qualys.com:443
- dist01.sjdc01.qualys.com:443
- nochost.sjdc01.qualys.com:443
- scanservice1.qualys.com:443
- all in 64.39.96.0/20

Qualys user account

Authentication to your Qualys account with valid Qualys credentials is required for making Qualys API requests to the Qualys API servers.

The application must authenticate using Qualys account credentials (user name and password) as part of the HTTP request. The credentials are transmitted using the “Basic Authentication Scheme” over HTTPS.

For information, see the “Basic Authentication Scheme” section of RFC #2617:

http://www.faqs.org/rfcs/rfc2617.html

The exact method of implementing authentication will vary according to which programming language is used.

The allowed methods, POST and/or GET, for each API request are documented with each API call in this user guide.

Sample request - basic authentication

```
curl -u "USERNAME:PASSWORD"
https://qualysapi.qualys.com/qps/rest/2.0/count/waf/webapp
```

Making API Calls

Curl samples in our API doc

We use curl in our API documentation to show an example how to form REST API calls, and it is not meant to be an actual production example of implementation.

Making Requests with an XML Payload

While it is still possible to create simple API requests using the GET method, you can create API requests using the POST method with an XML payload to make an advanced request.

The XML payloads can be compared to a scripting language that allows user to make multiple actions within one single API request, like adding a parameter to an object and updating another parameter.

The XML structure of the payload is described in the XSD files.

XML Output Pagination / Truncation

The XML output of a search API request is paginated and the default page size is 100 object records. The page size can be customized to a value between 1 and 1,000. If the number of records is greater than the page size then the <ServiceResponse> element shows the response code SUCCESS with the element <hasMoreRecords>true</hasMoreRecords> as shown below.

Tutorial - Search clusters

This tutorial shows you how to use the pagination options. Search for clusters and limit results to 3 records.

API request

```
curl -u "USERNAME:PASSWORD" -X "POST" -H "Content-Type: text/xml"
https://qualysapi.qualys.com/qps/rest/2.0/search/waf/cluster --data <
file.xml

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

Request POST data

```
<ServiceRequest>
  <preferences>
    <startFromOffset>1</startFromOffset>
    <limitResults>3</limitResults>
  </preferences>
  <filters>
    <Criteria field=\"name\" operator=\"NOT EQUALS\">Demo
cluster</Criteria>
  </filters>
</ServiceRequest>
```

The number of records is greater than the default pagination value (100) so the <ServiceResponse> element identifies the last ID of the object in the current page output (i.e. lastId).

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.
0/waf/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
  <hasMoreRecords>true</hasMoreRecords>
  <lastId>233603</lastId>
  <data>
    <Cluster>
      <id>233601</id>
      <uuid>b8463eb1-ad9e-4151-9f9f-73d118b7aa71</uuid>
      <name><![CDATA[test-1232]]></name>
      <errorResponse>
        <customPage>
          <id>17803</id>
          <uuid>becfabfb-1afb-4d7e-b6dd-6884bcb35a52</uuid>
          <name><![CDATA[Copy of newpage]]></name>
        </customPage>
      </errorResponse>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,SAT,WED,SUN,THU,TUE,FRI</weekDays>
```

```
        <startTime>0</startTime>
        <timezone>
          <code>Asia/Colombo</code>
          <offset>+05:30</offset>
        </timezone>
        <nextUpgradeDate>2018-11-27T00:00:00Z</nextUpgradeDate>
      </updateSchedule>
      <owner>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </owner>
      <created>2018-07-12T11:05:35Z</created>
      <createdBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </createdBy>
      <updated>2018-07-12T11:06:10Z</updated>
      <updatedBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </updatedBy>
      <token><![CDATA[D22C8290-1D75-418D-B04E-4CC9903EC1FA]]></token>
      <syncDate>2018-11-26T10:20:17Z</syncDate>
      <status>NO_SENSORS</status>
      <deploymentStatus>UNUSED</deploymentStatus>
      <deployed>2018-07-24T11:03:39Z</deployed>
    </Cluster>
    <Cluster>
      <id>233602</id>
      <uuid>38b1bde1-4995-485c-9211-7151fcb53cc6</uuid>
      <name><![CDATA[test123214324]]></name>
      <errorResponse>
        <customPage>
          <id>20401</id>
          <uuid>8aa9df97-d0ad-4d56-862f-af0e62863412</uuid>
          <name><![CDATA[New-Custom-Page-John]]></name>
        </customPage>
      </errorResponse>
      <updateSchedule>
```

```
        <enabled>true</enabled>
        <weekDays>MON,TUE,WED,THU,FRI,SAT,SUN</weekDays>
        <startTime>0</startTime>
        <timezone>
          <code>Asia/Colombo</code>
          <offset>+05:30</offset>
        </timezone>
        <freezeEndDate>2018-08-30</freezeEndDate>
        <nextUpgradeDate>2018-11-27T00:00:00Z</nextUpgradeDate>
      </updateSchedule>
      <owner>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </owner>
      <created>2018-07-12T11:08:49Z</created>
      <createdBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </createdBy>
      <updated>2018-08-17T10:49:12Z</updated>
      <updatedBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </updatedBy>
      <token><![CDATA[1D0E6CCE-F436-40FB-8969-560585860E4E]]></token>
      <syncDate>2018-11-26T10:20:17Z</syncDate>
      <status>NO_SENSORS</status>
      <deploymentStatus>UNUSED</deploymentStatus>
      <deployed>2018-08-20T06:14:29Z</deployed>
    </Cluster>
    <Cluster>
      <id>233603</id>
      <uuid>e2d32eba-b45d-4ce2-951b-bf94978c0f57</uuid>
      <name><![CDATA[test@123]]></name>
      <errorResponse>
        <customPage>
          <id>17803</id>
          <uuid>becfabfb-1afb-4d7e-b6dd-6884bcb35a52</uuid>
          <name><![CDATA[Copy of newpage]]></name>
```

```
        </customPage>
      </errorResponse>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,SAT,WED,SUN,THU,TUE,FRI</weekDays>
        <startTime>0</startTime>
        <timezone>
          <code>Asia/Colombo</code>
          <offset>+05:30</offset>
        </timezone>
        <nextUpgradeDate>2018-11-27T00:00:00Z</nextUpgradeDate>
      </updateSchedule>
      <owner>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </owner>
      <created>2018-07-12T11:12:50Z</created>
      <createdBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </createdBy>
      <updated>2018-07-12T11:13:04Z</updated>
      <updatedBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </updatedBy>
      <token><![CDATA[677D4FE8-7AC9-4975-A60D-7C9C209085B1]]></token>
      <syncDate>2018-11-26T10:20:18Z</syncDate>
      <status>NO_SENSORS</status>
      <deploymentStatus>UNUSED</deploymentStatus>
      <deployed>2018-07-24T11:03:39Z</deployed>
    </Cluster>
  </data>
</ServiceResponse>
```

To get the next page of results, you need to edit your service request in the data section that will be passed to API request as a POST payload. The next

page of results are according to the <lastId> element returned in the first page.

API request

```
curl -u "USERNAME:PASSWORD" -X "POST" -H "Content-Type: text/xml"
https://qualysapi.qualys.com/qps/rest/2.0/search/waf/cluster --data <
file.xml

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

Request POST data

```
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<ServiceRequest>
  <preferences>
  <startFromOffset>4</startFromOffset>
  <limitResults>3</limitResults>
  </preferences>
  <filters>
  <Criteria field=\"name" operator=\"NOT EQUALS\">Demo
cluster</Criteria>
 </filters>
</ServiceRequest>
```

The number of records is greater than the default pagination value (100) so the <ServiceResponse> element identifies the last ID of the object in the current page output (i.e. lastId).

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.
0/waf/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
  <hasMoreRecords>true</hasMoreRecords>
  <lastId>234606</lastId>
  <data>
    <Cluster>
      <id>233604</id>
      <uuid>1a70b8b3-bdb7-4de5-b36a-f576615e7295</uuid>
```

```
      <name><![CDATA[test@1234]]></name>
      <errorResponse>
        <customPage>
          <id>17803</id>
          <uuid>becfabfb-1afb-4d7e-b6dd-6884bcb35a52</uuid>
          <name><![CDATA[Copy of newpage]]></name>
        </customPage>
      </errorResponse>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,SAT,WED,SUN,THU,TUE,FRI</weekDays>
        <startTime>0</startTime>
        <timezone>
          <code>Asia/Colombo</code>
          <offset>+05:30</offset>
        </timezone>
        <nextUpgradeDate>2018-11-27T00:00:00Z</nextUpgradeDate>
      </updateSchedule>
      <owner>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </owner>
      <created>2018-07-12T11:14:25Z</created>
      <createdBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </createdBy>
      <updated>2018-07-12T11:15:05Z</updated>
      <updatedBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </updatedBy>
      <token><![CDATA[27B2B72F-4D64-4943-8587-A4CB67CB5CEF]]></token>
      <syncDate>2018-11-26T10:22:18Z</syncDate>
      <status>NO_SENSORS</status>
      <deploymentStatus>UNUSED</deploymentStatus>
      <deployed>2018-07-24T11:03:39Z</deployed>
    </Cluster>
    <Cluster>
```

```
      <id>233605</id>
      <uuid>55999918-508d-46ac-8f63-26102a7d994d</uuid>
      <name><![CDATA[test-123]]></name>
      <errorResponse>
        <customPage>
          <id>17803</id>
          <uuid>becfabfb-1afb-4d7e-b6dd-6884bcb35a52</uuid>
          <name><![CDATA[Copy of newpage]]></name>
        </customPage>
      </errorResponse>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,TUE,WED,THU,FRI,SAT,SUN</weekDays>
        <startTime>0</startTime>
        <timezone>
          <code>Asia/Colombo</code>
          <offset>+05:30</offset>
        </timezone>
        <nextUpgradeDate>2018-11-27T00:00:00Z</nextUpgradeDate>
      </updateSchedule>
      <owner>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </owner>
      <created>2018-07-12T11:23:22Z</created>
      <createdBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </createdBy>
      <updated>2018-07-12T11:23:22Z</updated>
      <updatedBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </updatedBy>
      <token><![CDATA[54F94C90-8167-4607-BC1E-68908BD60828]]></token>
      <syncDate>2018-11-26T10:22:18Z</syncDate>
      <status>NO_SENSORS</status>
      <deploymentStatus>UNUSED</deploymentStatus>
      <deployed>2018-08-20T06:14:29Z</deployed>
```

```
    </Cluster>
    <Cluster>
      <id>233606</id>
      <uuid>2b8e6a70-4b5b-4dea-b660-d4a095ef45d6</uuid>
      <name><![CDATA[new-cluster]]></name>
      <errorResponse>
        <customPage>
          <id>17803</id>
          <uuid>becfabfb-1afb-4d7e-b6dd-6884bcb35a52</uuid>
          <name><![CDATA[Copy of newpage]]></name>
        </customPage>
      </errorResponse>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,SAT,WED,SUN,THU,TUE,FRI</weekDays>
        <startTime>0</startTime>
        <timezone>
          <code>Asia/Colombo</code>
          <offset>+05:30</offset>
        </timezone>
        <nextUpgradeDate>2018-11-27T00:00:00Z</nextUpgradeDate>
      </updateSchedule>
      <owner>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </owner>
      <created>2018-07-12T11:37:16Z</created>
      <createdBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </createdBy>
      <updated>2018-07-12T11:38:07Z</updated>
      <updatedBy>
        <id>10141299</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </updatedBy>
      <token><![CDATA[3A6E9002-E791-4EBA-A068-982DAE13EF80]]></token>
      <syncDate>2018-11-26T10:22:17Z</syncDate>
      <status>NO_SENSORS</status>
```

```
      <deploymentStatus>UNUSED</deploymentStatus>
      <deployed>2018-08-20T06:14:29Z</deployed>
    </Cluster>
  </data>
</ServiceResponse>
```

Setting custom page size

The service request needs to contain the <preferences> section with the <limitResults> parameter. For the <limitResults> parameter you can enter a value from 1 to 1,000. You can change which objects are returned and the number of objects by specifying a preferences tag in the POST body of your request.

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <preferences>
    <startFromOffset>4</startFromOffset>
    <limitResults>3</limitResults>
  </preferences>
  <filters>
    <Criteria field="name" operator="NOT EQUALS">Demo
cluster</Criteria>
  </filters>
</ServiceRequest>
```

Preferences tag fields:

startFromOffset - The first item to return by index. The default is 1.

startFromId - The first item to return by primary key. No default value.

limitResults - The total number of items to return. The default is 100.

Supported operators

These filter operators are supported for use with <ServiceRequest> parameters. Keep in mind many API calls support <ServiceRequest> parameters, and some API calls do not support <ServiceRequest> parameters. Please see the description for each API call for details.

Long - EQUALS, NOT EQUALS

UUID - EQUALS, NOT EQUALS

Text - CONTAINS, EQUALS, NOT EQUALS

Integer - EQUALS, NOT EQUALS, GREATER, LESSER, IN

Float - EQUALS, IN, NOT EQUALS, GREATER, LESSER

Date - EQUALS, NOT EQUALS, GREATER, LESSER

Keyword - EQUALS, NOT EQUALS, IN

Boolean - (true/false) EQUALS, NOT EQUALS

Important Notes

1) For searching appliances, the elements systemRam, systemCpusCount, and systemCpusCores support EQUALS, IN, NOT EQUALS, GREATER, LESSER. The element ip supports only EQUALS operator.

2) For searching clusters, the elements createdBy.id and updatedBy.id support only EQUALS operator.

3) For searching events,

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

- The element qids supports only IN operator.

- The element webApplication supports only CONTAINS operator.

- The element "occurred" support only EQUALS, GREATER and LESSER operators.

Tracking API usage by user

You can track API usage per user without the need to provide user credentials such as the username and password. Contact Qualys Support to get the X-Powered-By HTTP header enabled.

Once enabled, the X-Powered-By HTTP header is returned for each API request made by a user. The X-Powered-By value includes a unique ID generated for each subscription and a unique ID generated for each user.

Optional X-Powered-By header

API usage can be tracked using the X-Powered-By HTTP header which includes a unique ID generated for each subscription and a unique ID generated for each user. Once enabled, the X-Powered-By HTTP header is returned for each API request made by a user. The X-Powered-By HTTP header will be returned for both valid and invalid requests. However, it will not be returned if an invalid URL is hit or when user authentication fails.

The X-Powered-By header is returned in the following format:

```
X-Powered-By: Qualys:<POD_ID>:<SUB_UUID>:<USER_UUID>
```

where,

- POD_ID is the shared POD or a PCP. Shared POD is USPOD1, USPOD2, etc.

- SUB_UUID is the unique ID generated for the subscription

- USER_UUID is the unique ID generated for the user. You can use the USER_UUID to track API usage per user.

Sample X-Powered-By header

```
X-Powered-By: Qualys:QAPOD4SJC:f972e2cc-69d6-7ebd-80e6-
7b9a931475d8:06198167-43f3-7591-802a-1c400a0e81b1
```

Know your portal version

/qps/rest/portal/version/

[GET] [POST]

Using the Version API you can find out the installed version of Portal and its sub-modules that are available in your subscription.

Sample XML

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Accept: application/xml"
https://qualysapi.qualys.com/qps/rest/portal/version
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/ve
rsion.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Portal-Version>
            <PortalApplication-VERSION>2.33.0.0-SNAPSHOT-1 DEVELOP
#352 (2018-05-07T22:53:43Z)</PortalApplication-VERSION>
            <WAS-VERSION>6.0.0.0</WAS-VERSION>
            <FIM-VERSION>1.5.1</FIM-VERSION>
            <VM-VERSION>1.0.3</VM-VERSION>
            <CERTVIEW-VERSION>1.1.0.0</CERTVIEW-VERSION>
            <CM-VERSION>1.20.1</CM-VERSION>
            <MDS-VERSION>2.11.7.0</MDS-VERSION>
            <CA-VERSION>2.9.1.0</CA-VERSION>
            <IOC-VERSION>1.1.0</IOC-VERSION>
            <AV2-VERSION>0.1.0</AV2-VERSION>
            <QUESTIONNAIRE-VERSION>2.14.0.4</QUESTIONNAIRE-VERSION>
            <WAF-VERSION>2.7.0.0</WAF-VERSION>
        </Portal-Version>
    </data>
```

```
</ServiceResponse>
```

Sample JSON

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Accept: application/json"
https://qualysapi.qualys.com/qps/rest/portal/version
```

Response

```
{
  "ServiceResponse": {
    "data": [
      {
        "Portal-Version": {
          "PortalApplication-VERSION": "2.33.0.0-SNAPSHOT-1 DEVELOP
#352 (2018-05-07T22:53:43Z)",
          "WAS-VERSION": "6.0.0.0",
          "VM-VERSION": "1.0.3",
          "CM-VERSION": "1.20.1",
          "MDS-VERSION": "2.11.7.0",
          "CA-VERSION": "2.9.1.0",
          "QUESTIONNAIRE-VERSION": "2.14.0.4",
          "WAF-VERSION": "2.7.0.0"
        },
...
                }
      }
    ],
    "responseCode": "SUCCESS",
    "count": 1
  }
}
```

Web Applications API

Current web application count

/qps/rest/2.0/count/waf/webapp/

[GET]

Returns the total number of web applications licensed for WAF and in the user's account.

Permissions 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

No input elements are available.

Sample

API request

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

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/
xsd/2.0/waf/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>72</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webapp.xsd

Get details on a web application

/qps/rest/2.0/get/waf/webapp/<id>

[GET]

Returns details about a specific web application licensed for WAF, within the user's scope. Want to find a web application ID to use as input? See Search web applications.

Permissions 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

The element "id" (Integer) is required, where "id" identifies the web application ID of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/get/waf/webapp/6662076"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.
0/waf/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>6662076</id>
            <uuid>943244ca-e1cb-415b-92a7-ebf115fc2988</uuid>
            <name>
                <![CDATA[Site 01]]>
            </name>
```

```
            <owner>
                <id>11826614</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2018-01-04T01:17:23Z</created>
            <createdBy>
                <id>11826614</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2018-07-18T22:54:19Z</updated>
            <updatedBy>
                <id>11826614</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <url>https://developers.google.com/analytics/devguides/col
lection/analyticsjs/cookies-user-id</url>
            <urls>
                <string>http://qcum123.com</string>
            </urls>
            <webServer>
                <id>83007</id>
                <uuid>9f7a4676-32f9-4290-9782-e4138c989d90</uuid>
                <name>
                    <![CDATA[Copy of waf-site711]]>
                </name>
            </webServer>
            <webServerTimeout>60</webServerTimeout>
            <persistencyEnabled>false</persistencyEnabled>
            <healthcheck>
                <id>40601</id>
                <uuid>e48247e5-84f8-4252-9a5f-22bac8cd0257</uuid>
                <name>
                    <![CDATA[Standard Healthcheck]]>
                </name>
            </healthcheck>
            <failureResponseCode>503</failureResponseCode>
            <certificate>
                <id>92401</id>
                <uuid>67a52056-dd4f-4644-bbdb-961e5960eebe</uuid>
```

```
                <name>
                    <![CDATA[ssl2.33]]>
                </name>
            </certificate>
            <sslProtocols>
                <![CDATA[TLS12]]>
            </sslProtocols>
            <sslSecurityFilters>
                <![CDATA[Strong,Weak]]>
            </sslSecurityFilters>
            <sslCiphers>
                <![CDATA[ADH-AES128-GCM-SHA256,ADH-AES128-
SHA256,ECDHE-RSA-AES256-SHA384]]>
            </sslCiphers>
            <blockingMode>true</blockingMode>
            <securityPolicy>
                <id>148003</id>
                <uuid>f99cdce6-0c1e-4814-8374-5e1595c9d7c1</uuid>
                <name>
                    <![CDATA[Copy of portal2.30Sanity]]>
                </name>
            </securityPolicy>
            <httpProfile>
                <id>48001</id>
                <uuid>bde48f0d-883e-4635-b171-cec1a9bea021</uuid>
                <name>
                    <![CDATA[WAFUI-1937]]>
                </name>
            </httpProfile>
            <scanTrustEnabled>false</scanTrustEnabled>
            <clusters>
                <Cluster>
                    <id>153801</id>
                    <uuid>1a487126-7eae-4910-b538-b0264343f8bd</uuid>
                    <name>
                        <![CDATA[qwaf06.p04.sjc01.eng.qualys.com]]>
                    </name>
                </Cluster>
            </clusters>
            <status>UNUSED</status>
            <sslEnabled>true</sslEnabled>
            <sslStatus>INVALID</sslStatus>
            <deploymentStatus>UNUSED</deploymentStatus>
            <deployed>2018-07-18T22:54:18Z</deployed>
        </WebApp>
```

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

XSD

<platform API server>/qps/xsd/2.0/waf/webapp.xsd

Search web applications

/qps/rest/2.0/search/waf/webapp

[POST]

Finds web applications in the user's account matching search criteria.

Permissions 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. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Supported filter operators

Parameter	Description
id (Long)	Web application identifier on Qualys Cloud Platform.
uuid (UUID)	Web application identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the Web application as defined by a user. This is unique in subscription. Valid action: Update
url (Text)	The incoming URL used. Any requests received that match that URL should be routed to this web application. Valid action: Update

tags.tag.id (Long)	The identifier of one tag associated with this WebApp asset.
tags.tag.name (Text)	The name of one tag associated with this WebApp asset.
owner.id (Long)	The user ID of the Web application owner.
owner.username (Text)	The user name of the Web application owner.
owner.firstname (Text)	The first name of the Web application owner.
owner.lastname (Text)	The last name of the Web application owner.
created (Date)	Date when the WAF server processed the Heartbeat message (stored and available). It is not updated directly, but retrieved asynchronously and is synchronized when Qualys Cloud platform/WAF module updates their respective records. The Heartbeat message contains system information such as systemOs, systemRam, and systemType.
createdBy.id (Long)	The user ID who created the Web application.
createdBy.username (Text)	The user name who created the Web application.
createdBy.firstname (Text)	The first name of the user who created the Web application.
createdBy.lastname (Text)	The last name of the user who created the Web application.
updated (Date)	The date/time when the Web application was last updated.

updatedBy.id (Long)	The user ID who last updated the Web application.
updatedBy.username (Text)	The user name who last updated the Web application.
updatedBy.firstname (Text)	The first name of the user who updated the Web application.
updatedBy.lastname (Text)	The last name of the user who updated the Web application.
urls.value (Text)	The type of CPU model on the WAF appliance. Simple text identifier that describes CPU virtualized (Such as "Intel Bi-Xeon xxxx")
healthcheck.id (Long)	The ID of the healthcheck assigned to the web application.
healthcheck.uuid (UUID)	The UUID of the healthcheck assigned to the web application.
healthcheck.name (Text)	The name of the healthcheck assigned to the web application.
failureResponseCode (Long)	Specify the response code returned when all Web servers in the server pool are down. The default value is 503. For example, a 503 page is displayed when the Web servers are down or the Web site is not reachable.
webServer.id (Long)	The web server pool where received requests should be routed to.
webServer.uuid (UUID)	The UUID of the web server assigned to the web application.
webServer.name (Text)	The name of the web server pool assigned to the web application.

webServerTimeout (Long)	Server Timeout is the maximum time to wait for an HTTP connection attempt to a server to succeed. If the HTTP request does not respond before the duration set, it will timeout and return an HTTP 503 error code. Specify a timeout period between 1 second to 3600 seconds. Default value is 60 seconds.
persistencyEnabled (Boolean)	Persistency allows the client to reconnect to the same server previously visited for the web application. This bypasses load balancing. Specify the cookie name to persist connection to the server previously visited by the client.
persistencyToken (Text)	The cookie name used to maintain sessions on the WebServer, if persistencyEnabled is true.
scanTrustEnabled (Boolean)	Enable scanTrust (Authenticated Scanning) for integration with Qualys WAS for vulnerability scanning. You must get this feature enabled in subscription before you can use it.
certificate.id (Long)	The ID of the SSL certificate assigned to the web application.
certificate.uuid (UUID)	The UUID of the SSL certificate assigned to the web application.
certificate.name (Text)	The name of the SSL certificate assigned to the web application.
status	(Text) Status information: Up - All servers assigned to the Web application are up and running. Down - All servers assigned to the Web application

	are down. Degraded - Atleast one server assigned to the Web application is down. Unused - Web application is not deployed on any WAF cluster, or the Web application is deployed on a cluster having no appliances registered to it.
deploymentStatus	(Text) The status of Web Application and other configurations being pushed to the cluster.: SUCCESS (Web application successfully deployed) PENDING_DEPLOY (Deployment of a Web application is pending) FAILURE (Deployment of Web application configuration has failed on all associated clusters) PARTIAL (Deployment failed on at least one cluster or appliance assigned to the Web application) IN_PROGRESS (Deployment of Web application on cluster is in progress) UNUSED (Web application is not deployed on any WAF cluster or the Web application is deployed on a cluster having no appliances registered to it)
deployed (Date)	The latest date on which the Web Application was pushed to the cluster.
synced (Date)	The date on which the Web application last synchronized with the cluster.
blockingMode	If enabled blocks incoming traffic and displays the

(Boolean)	default WAF error page when it violates the selected security policy. If customPage is provided, displays the custom response page when incoming traffic is blocked. If not enabled, only monitors (logs) the incoming traffic.
customPage.id (Long)	The ID of the custom response page assigned to the web application.
customPage.uuid (UUID)	The UUID of the custom response page assigned to the web application.
customPage.name (Text)	The name of the custom response page assigned to the web application.
securityPolicy.id (Long)	The ID of the security policy assigned to the Web application.
securityPolicy.uuid (UUID)	The UUID of the security policy assigned to the Web application.
securityPolicy.name (Text)	The name of the security policy created by a user, assigned to the Web application.
httpProfile.id (Long)	The ID of the HTTP profile assigned to the web application.
httpProfile.uuid (UUID)	The UUID of the HTTP profile assigned to the web application.
httpProfile.name (Text)	The name of the HTTP profile assigned to the web application.
sslEnabled (Boolean)	Is SSL enabled? TRUE if any incoming URL declared in 'url' or 'urls.string' are using HTTPS.
clusters.cluster.id (Long)	A WAF cluster ID used to deploy the Web application.

clusters.cluster.name (Text)	The name of a WAF cluster, created by a user, used to deploy the Web application.
clusters.cluster.uuid (UUID)	A UUID for a WAF cluster used to deploy the Web application.

Web applications API only provides the id, uuid, and name of a cluster. Use the Clusters API to get more details of a cluster and its appliances.

Sample

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/webapp"

Note: "file.xml" contains the request POST data.
The request POST data is optional. If you leave it empty all web
applications in the user's scope are returned.
```

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
     <ServiceRequest>
       <filters>
         <Criteria field="owner.id"
operator="EQUALS">228723</Criteria>
         <Criteria field="certificate.name"
operator="CONTAINS">ssl2.33</Criteria>
       </filters>
     </ServiceRequest>
```

Response

```
<?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
/waf/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
```

```
    <data>
        <WebApp>
            <id>63098273</id>
            <uuid>01bd1b58-2802-48dd-b5b5-ea1342aea21a</uuid>
            <name>
                <![CDATA[Site 01]]>
            </name>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-05-31T09:01:49Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-05-31T09:19:39Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <url>https://site01.xfuentes-docker</url>
            <urls/>
            <webServer>
                <id>1001</id>
                <uuid>315cc797-3c73-4721-ba42-263e7e7b6cbb</uuid>
                <name>
                    <![CDATA[First Pool]]>
                </name>
            </webServer>
            <webServerTimeout>60</webServerTimeout>
            <persistencyEnabled>false</persistencyEnabled>
            <healthcheck>
                <id>1001</id>
                <uuid>f479e6f5-57a1-4677-a8cb-272e2c69623a</uuid>
                <name>
                    <![CDATA[Standard Healthcheck]]>
                </name>
            </healthcheck>
```

```
            <failureResponseCode>503</failureResponseCode>
            <certificate>
                <id>1</id>
                <uuid>a21b4a1b-de54-45e8-9d29-204444cef5bb</uuid>
                <name>
                    <![CDATA[ssl2.33]]
                </name>
            </certificate>
            <sslProtocols>
                <![CDATA[TLS12]]>
            </sslProtocols>
            <sslSecurityFilters>
                <![CDATA[Strong,Weak]]>
            </sslSecurityFilters>
            <sslCiphers>
                <![CDATA[ADH-AES128-GCM-SHA256,ADH-AES128-
SHA256,ECDHE-RSA-AES256-SHA384]]>
            </sslCiphers>
            <blockingMode>false</blockingMode>
            <customPage>
                <id>1001</id>
                <uuid>0dba4434-1118-40e5-8768-23c5616053d5</uuid>
                <name>
                    <![CDATA[My Response]]>
                </name>
            </customPage>
            <securityPolicy>
                <id>30682</id>
                <uuid>6c56416a-66ff-4016-b16f-da2cec2e97f3</uuid>
                <name>
                    <![CDATA[Standard Policy]]>
                </name>
            </securityPolicy>
            <httpProfile>
                <id>1001</id>
                <uuid>341bcf25-c9fa-45ff-ac63-728e38056443</uuid>
                <name>
                    <![CDATA[Standard Protocol]]>
                </name>
            </httpProfile>
            <scanTrustEnabled>false</scanTrustEnabled>
            <clusters>
            <customRules>
                <CustomRule>
                    <id>1001</id>
```

```
                    <uuid>20e220d3-1244-42ca-a473-c80469e95bc0</uuid>
                    <name>
                        <![CDATA[Test custom rule]]>
                    </name>
                </CustomRule>
                <CustomRule>
                    <id>2001</id>
                    <uuid>c64c3008-c1af-4969-8290-d0b1d8e9f27b</uuid>
                    <name>
                        <![CDATA[shamzor]]>
                    </name>
                </CustomRule>
            </customRules>
            <clusters>
                <Cluster>
                    <id>24401</id>
                    <uuid>48ae444d-e652-443f-8438-3a9182403b9f</uuid>
                    <name>
                        <![CDATA[Cluster 1]]>
                    </name>
                </Cluster>
            </clusters>
            <status>DOWN</status>
            <sslEnabled>true</sslEnabled>
            <sslStatus>OK</sslStatus>
            <deploymentStatus>FAILURE</deploymentStatus>
            <deployed>2017-05-31T12:15:14Z</deployed>
        </WebApp>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webapp.xsd

Create web application

/qps/rest/2.0/create/waf/webapp

[POST]

Create a web application asset that you want to monitor using WAF.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Create WAF Asset permissions.

Input Parameters

Allowed input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	(Required) The name of the Web application as defined by a user. This is unique in subscription. Valid action: Update
url (Text)	(Required) The incoming URL used. Any requests received that match that URL should be routed to this web application. Valid action: Update
webServer.id (Long)	(Required) The web server pool where received requests should be routed to.
securityPolicy.id (Long)	(Required) The ID of the security policy assigned to the Web application.
httpProfile.id (Long)	(Required) The ID of the HTTP profile

	assigned to the web application.
persistencyEnabled (Boolean)	Persistency allows the client to reconnect to the same server previously visited for the web application. This bypasses load balancing. Specify the cookie name to persist connection to the server previously visited by the client.
persistencyToken (Text)	The cookie name used to maintain sessions on the WebServer, if persistencyEnabled is true.
healthcheck.id (Long)	The ID of the healthcheck assigned to the web application.
failureResponseCode (Long)	Specify the response code returned when all Web servers in the server pool are down. The default value is 503. For example, a 503 page is displayed when the Web servers are down or the Web site is not reachable.
certificate.id (Long)	The ID of the SSL certificate assigned to the web application.
sslProtocols (Text)	A comma separated list of allowed SSL protocols (SSLV3,TLS10,TLS11,TLS12,TLS13) Default protocols are TLS11,TLS12,TLS13
sslSecurityFilters (Text)	A comma separated list of allowed SSL security filters (Strong, Good, Weak, Unsafe) Default security filters are Strong, Good
sslCiphers (Text)	A comma separated list of allowed SSL ciphers (ECDHERSA-AES256-GCM-SHA384,ECDHE-ECDSA-AES256-

	GCMSHA384, ECDHE-RSA-AES256-SHA384 etc...)
blockingMode (Boolean)	If enabled blocks incoming traffic and displays the default WAF error page when it violates the selected security policy. If customPage is provided, displays the custom response page when incoming traffic is blocked. If not enabled, only monitors (logs) the incoming traffic.
customPage.id (Long)	The ID of the custom response page assigned to the web application.
scanTrustEnabled (Boolean)	Enable scanTrust (Authenticated Scanning) for integration with Qualys WAS for vulnerability scanning. You must get this feature enabled in subscription before you can use it.
customRules.CustomRule.id (Long)	The ID of the custom rule assigned to the web application.
clusters.cluster.id (Long)	A WAF cluster ID used to deploy the Web application.
lastComment (Text)	The last user defined comment.
urls	The list of optional (aliases) URL for this web application (incoming URL)
urls.string (Text)	One aliases URL for this web application (incoming URL)
tags.tag.id (Long)	The identifier of one tag associated with this WebApp asset.
tags.tag.name (Text)	The name of one tag associated with this WebApp asset.

webServerTimeout (Long)	Server Timeout is the maximum time to wait for an HTTP connection attempt to a server to succeed. If the HTTP request does not respond before the duration set, it will timeout and return an HTTP 503 error code. Specify a timeout period between 1 second to 3600 seconds. Default value is 60 seconds.

Sample

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/webapp"

Note: "file.xml" contains the request POST data.
The request POST data is optional. If you leave it empty all web
applications in the user’s scope are returned.
```

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
   <data>
      <WebApp>
         <name>Sslsite1by post</name>
         <url>https://sslsite1.com</url>
         <webServer>
            <id>83002</id>
         </webServer>
         <webServerTimeout>3600</webServerTimeout>
         <healthcheck>
           <id>122004</id>
           </healthcheck>
           <failureResponseCode>503</failureResponseCode>
           <certificate>
                <id>92401</id>
            </certificate>
            <sslProtocols>TLS12</sslProtocols>
            <sslSecurityFilters>Strong,Weak</sslSecurityFilters>
```

```
          <sslCiphers> ADH-AES128-GCM-SHA256,ADH-AES128-SHA256,ECDHE-
RSA-AES256-SHA384</sslCiphers>
         <blockingMode>true</blockingMode>
         <securityPolicy>
            <id>148003</id>
         </securityPolicy>
         <httpProfile>
            <id>48001</id>
         </httpProfile>
         <clusters>
                <Cluster>
                    <id>153801</id>
                </Cluster>
          </clusters>
      </WebApp>
   </data>
</ServiceRequest>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.
0/waf/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>7831329</id>
            <uuid>ed13870b-66c6-4ba8-8cdd-66aea6c20c36</uuid>
            <name>
                <![CDATA[Site created by API]]>
            </name>
            <owner>
                <id>11826614</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2018-07-20T04:58:18Z</created>
            <createdBy>
                <id>11826614</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
```

```
            </createdBy>
            <updated>2018-07-20T04:58:18Z</updated>
            <updatedBy>
                <id>11826614</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <url>https://sslsite1.com</url>
            <webServer>
                <id>83002</id>
                <uuid>48455b2c-6456-467b-ba5f-02ab9e0bc7fc</uuid>
                <name>
                    <![CDATA[Copy of wafsite6-Two-2]]>
                </name>
            </webServer>
            <webServerTimeout>3600</webServerTimeout>
            <persistencyEnabled>false</persistencyEnabled>
            <healthcheck>
                <id>122004</id>
                <uuid>bbc4d877-1c1f-4446-8395-7fdadc32a467</uuid>
                <name>
                    <![CDATA[Copy --of Standard Healthcheck]]>
                </name>
            </healthcheck>
            <failureResponseCode>503</failureResponseCode>
            <certificate>
                <id>92401</id>
                <uuid>67a52056-dd4f-4644-bbdb-961e5960eebe</uuid>
                <name>
                    <![CDATA[ssl2.33]]>
                </name>
            </certificate>
            <sslProtocols>
                <![CDATA[TLS12]]>
            </sslProtocols>
            <sslSecurityFilters>
                <![CDATA[Strong,Weak]]>
            </sslSecurityFilters>
            <sslCiphers>
                <![CDATA[ADH-AES128-GCM-SHA256,ADH-AES128-
SHA256,ECDHE-RSA-AES256-SHA384]]>
            </sslCiphers>
            <blockingMode>true</blockingMode>
            <securityPolicy>
```

```
                <id>148003</id>
                <uuid>f99cdce6-0c1e-4814-8374-5e1595c9d7c1</uuid>
                <name>
                    <![CDATA[Copy of portal2.30Sanity]]>
                </name>
            </securityPolicy>
            <httpProfile>
                <id>48001</id>
                <uuid>bde48f0d-883e-4635-b171-cec1a9bea021</uuid>
                <name>
                    <![CDATA[WAFUI-1937]]>
                </name>
            </httpProfile>
            <scanTrustEnabled>false</scanTrustEnabled>
            <clusters>
                <Cluster>
                    <id>153801</id>
                    <uuid>1a487126-7eae-4910-b538-b0264343f8bd</uuid>
                    <name>
                        <![CDATA[qwaf06.p04.sjc01.eng.qualys.com]]>
                    </name>
                </Cluster>
            </clusters>
            <status>DOWN</status>
            <sslEnabled>true</sslEnabled>
            <sslStatus>INVALID</sslStatus>
            <deploymentStatus>PENDING_DEPLOY</deploymentStatus>
            <deployed>2018-07-20T04:58:18Z</deployed>
        </WebApp>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webapp.xsd

Update web application

/qps/rest/2.0/update/waf/webapp/<id>

[POST]

Update a web application asset in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Update WAF Asset permissions, and web apps licensed for WAF and within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the web application you want to update.

Optional input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	The name of the Web application as defined by a user. This is unique in subscription. Valid action: Update
certificate.id (Long)	The ID of the SSL certificate assigned to the web application (if ssl is enabled).
url (Text)	The incoming URL used. Any requests received that match that URL should be routed to this web application. Valid action: Update

webServer.id (Long)	The web server pool where received requests should be routed to.
webServerTimeout (Long)	Server Timeout is the maximum time to wait for an HTTP connection attempt to a server to succeed. If the HTTP request does not respond before the duration set, it will timeout and return an HTTP 503 error code. Specify a timeout period between 1 second to 3600 seconds. Default value is 60 seconds.
securityPolicy.id (Long)	The ID of the security policy assigned to the Web application.
httpProfile.id (Long)	The ID of the HTTP profile assigned to the web application.
persistencyEnabled (Boolean)	Persistency allows the client to reconnect to the same server previously visited for the web application. This bypasses load balancing. Specify the cookie name to persist connection to the server previously visited by the client.
persistencyToken (Text)	The cookie name used to maintain sessions on the WebServer, if persistencyEnabled is true.
healthcheck.id (Long)	The ID of the healthcheck assigned to the web application.
failureResponseCode (Long)	Specify the response code returned when all Web servers in the server pool are down. The default value is 503. For example, a 503 page is displayed when the Web servers are down or the Web site is not reachable.

certificate.id (Long)	The ID of the SSL certificate assigned to the web application.
sslProtocols (Text)	A comma separated list of allowed SSL protocols (SSLV3,TLS10,TLS11,TLS12,TLS13) Default protocols are TLS11,TLS12,TLS13
sslSecurityFilters (Text)	A comma separated list of allowed SSL security filters (Strong, Good, Weak, Unsafe) Default security filters are Strong, Good
sslCiphers (Text)	A comma separated list of allowed SSL ciphers (ECDHERSA-AES256-GCM-SHA384,ECDHE-ECDSA-AES256-GCMSHA384, ECDHE-RSA-AES256-SHA384 etc...)
blockingMode (Boolean)	If enabled blocks incoming traffic and displays the default WAF error page when it violates the selected security policy. If customPage is provided, displays the custom response page when incoming traffic is blocked. If not enabled, only monitors (logs) the incoming traffic.
customPage.id (Long)	The ID of the custom response page assigned to the web application.
scanTrustEnabled (Boolean)	Enable scanTrust (Authenticated Scanning) for integration with Qualys WAS for vulnerability scanning. You must get this feature enabled in subscription before you can use it.
customRules.CustomRule.id (Long)	The ID of the custom rule assigned to the web application.

clusters.cluster.id (Long)	A WAF cluster ID used to deploy the Web application.
lastComment (Text)	The last user defined comment.
urls (Text)	The list of optional (aliases) URL for this web application (incoming URL)
urls.string (Text)	One aliases URL for this web application (incoming URL)
tags	The list of tags associated with that WebApp asset.

Sample

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/webapp/63098473"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <WebApp>
            <url>https://site02.xfuentes-docker</url>
            <certificate><id>1001</id></certificate>
            <sslProtocols>TLS12</sslProtocols>
            <sslSecurityFilters>Good</sslSecurityFilters>
            <blockingMode>true</blockingMode>
            <customRules><CustomRule><id>2001</id></CustomRule></custo
mRules>
        </WebApp>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>63098473</id>
            <uuid>6ecd55d8-5431-4114-ba11-90b020576f37</uuid>
            <name>
                <![CDATA[Site created by API]]>
            </name>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-06-01T09:22:47Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-06-02T12:23:32Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <url>https://site02.xfuentes-docker</url>
            <webServer>
                <id>1001</id>
                <uuid>315cc797-3c73-4721-ba42-263e7e7b6cbb</uuid>
                <name>
                    <![CDATA[First Pool]]>
                </name>
            </webServer>
            <webServerTimeout>60</webServerTimeout>
            <persistencyEnabled>true</persistencyEnabled>
            <persistencyToken>
```

```
                <![CDATA[ptoken]]>
            </persistencyToken>
            <healthcheck>
                <id>1001</id>
                <uuid>f479e6f5-57a1-4677-a8cb-272e2c69623a</uuid>
                <name>
                    <![CDATA[Standard Healthcheck]]>
                </name>
            </healthcheck>
            <certificate>
                <id>1001</id>
                <uuid>5788c0eb-5bda-466f-bfb6-71a1f60856ff</uuid>
                <name>
                    <![CDATA[Site02 Cert]]>
                </name>
            </certificate>
            <sslProtocols>
                <![CDATA[TLS12]]>
            </sslProtocols>
            <sslSecurityFilters>
                <![CDATA[Good]]>
            </sslSecurityFilters>
            <sslCiphers>
                <![CDATA[ECDH-ECDSA-AES128-GCM-SHA256,ECDH-RSA-AES128-
SHA256,ECDH-RSA-AES128-GCM-SHA256,DHE-RSA-AES128-GCM-SHA256,ECDHE-
ECDSA-AES128-SHA256,DH-RSA-AES128-SHA256,DH-RSA-AES256-SHA256,ECDH-
ECDSA-AES128-SHA256,DHE-RSA-AES256-SHA256,DH-RSA-AES128-GCM-
SHA256,ECDHE-RSA-AES128-GCM-SHA256,ECDHE-RSA-AES128-SHA256,ECDHE-
ECDSA-AES128-GCM-SHA256,DHE-RSA-AES128-SHA256]]>
            </sslCiphers>
            <blockingMode>true</blockingMode>
            <securityPolicy>
                <id>30682</id>
                <uuid>6c56416a-66ff-4016-b16f-da2cec2e97f3</uuid>
                <name>
                    <![CDATA[Standard Policy]]>
                </name>
            </securityPolicy>
            <httpProfile>
                <id>1001</id>
                <uuid>341bcf25-c9fa-45ff-ac63-728e38056443</uuid>
                <name>
                    <![CDATA[Standard Protocol]]>
                </name>
            </httpProfile>
```

```
            <scanTrustEnabled>true</scanTrustEnabled>
            <scanTrustToken>
                <![CDATA[38770c30-7c79-4b75-a5ec-43d07493eca1]]>
            </scanTrustToken>
            <customRules>
                <CustomRule>
                    <id>2001</id>
                    <uuid>c64c3008-c1af-4969-8290-d0b1d8e9f27b</uuid>
                    <name>
                        <![CDATA[shamzor]]>
                    </name>
                </CustomRule>
            </customRules>
            <status>INACTIVE</status>
            <sslEnabled>true</sslEnabled>
            <sslStatus>OK</sslStatus>
            <deploymentStatus>PENDING_DEPLOY</deploymentStatus>
            <deployed>2017-06-01T16:05:54Z</deployed>
        </WebApp>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webapp.xsd

Update web applications (bulk)

/qps/rest/2.0/update/waf/webapp

[POST]

Update multiple web application assets in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Update WAF Asset permissions, and web apps licensed for WAF and within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search web applications.

Allowed input elements for bulk update are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Supported filter operators

Parameter	Description
name (Text)	The name of the Web application as defined by a user. This is unique in subscription. Valid action: Update
certificate.id (Long)	The ID of the SSL certificate assigned to the web application (if ssl is enabled).
url (Text)	The incoming URL used. Any requests received that match that URL should be

	routed to this web application. Valid action: Update
webServer.id (Long)	The web server pool where received requests should be routed to.
webServerTimeout (Long)	Server Timeout is the maximum time to wait for an HTTP connection attempt to a server to succeed. If the HTTP request does not respond before the duration set, it will timeout and return an HTTP 503 error code. Specify a timeout period between 1 second to 3600 seconds. Default value is 60 seconds.
securityPolicy.id (Long)	The ID of the security policy assigned to the Web application.
httpProfile.id (Long)	The ID of the HTTP profile assigned to the web application.
persistencyEnabled (Boolean)	Persistency allows the client to reconnect to the same server previously visited for the web application. This bypasses load balancing. Specify the cookie name to persist connection to the server previously visited by the client.
persistencyToken (Text)	The cookie name used to maintain sessions on the WebServer, if persistencyEnabled is true.
healthcheck.id (Long)	The ID of the healthcheck assigned to the web application.
failureResponseCode (Long)	Specify the response code returned when all

	Web servers in the server pool are down. The default value is 503. For example, a 503 page is displayed when the Web servers are down or the Web site is not reachable.
certificate.id (Long)	The ID of the SSL certificate assigned to the web application.
sslProtocols (Text)	A comma separated list of allowed SSL protocols (SSLV3,TLS10,TLS11,TLS12,TLS13) Default protocols are TLS11,TLS12,TLS13
sslSecurityFilters (Text)	A comma separated list of allowed SSL security filters (Strong, Good, Weak, Unsafe) Default security filters are Strong, Good
sslCiphers (Text)	A comma separated list of allowed SSL ciphers (ECDHERSA-AES256-GCM-SHA384,ECDHE-ECDSA-AES256-GCMSHA384, ECDHE-RSA-AES256-SHA384 etc...)
blockingMode (Boolean)	If enabled blocks incoming traffic and displays the default WAF error page when it violates the selected security policy. If customPage is provided, displays the custom response page when incoming traffic is blocked. If not enabled, only monitors (logs) the incoming traffic.
customPage.id (Long)	The ID of the custom response page assigned to the web application.
scanTrustEnabled (Boolean)	Enable scanTrust (Authenticated Scanning) for integration with Qualys WAS for vulnerability scanning. You must get this feature enabled in subscription before you

	can use it.
customRules.CustomRule.id (Long)	The ID of the custom rule assigned to the web application.
clusters.cluster.id (Long)	A WAF cluster ID used to deploy the Web application.
lastComment (Text)	The last user defined comment.
urls (Text)	The list of optional (aliases) URL for this web application (incoming URL)
urls.string (Text)	One aliases URL for this web application (incoming URL)
tags	The list of tags associated with that WebApp asset.

Sample

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/webapp"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <filters>
        <Criteria field="sslEnabled" operator="EQUALS">true</Criteria>
    </filters>
    <data>
        <WebApp>
            <sslProtocols>TLS12</sslProtocols>
            <sslSecurityFilters>Good</sslSecurityFilters>
           <customRules><CustomRule><id>2001</id></CustomRule></custom
Rules>
```

```
        </WebApp>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <WebApp>
            <id>63098273</id>
            <uuid>01bd1b58-2802-48dd-b5b5-ea1342aea21a</uuid>
            <name>
                <![CDATA[Site 01]]>
            </name>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-05-31T09:01:49Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-06-02T13:23:43Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <url>https://site01.xfuentes-docker</url>
            <webServer>
                <id>1001</id>
                <uuid>315cc797-3c73-4721-ba42-263e7e7b6cbb</uuid>
                <name>
                    <![CDATA[First Pool]]>
```

```
                </name>
            </webServer>
            <webServerTimeout>60</webServerTimeout>
            <persistencyEnabled>false</persistencyEnabled>
            <healthcheck>
                <id>1001</id>
                <uuid>f479e6f5-57a1-4677-a8cb-272e2c69623a</uuid>
                <name>
                    <![CDATA[Standard Healthcheck]]>
                </name>
            </healthcheck>
            <failureResponseCode>503</failureResponseCode>
            <certificate>
                <id>1</id>
                <uuid>a21b4a1b-de54-45e8-9d29-204444cef5bb</uuid>
                <name>
                    <![CDATA[Site01 Cert]]>
                </name>
            </certificate>
            <sslProtocols>
                <![CDATA[TLS12]]>
            </sslProtocols>
            <sslSecurityFilters>
                <![CDATA[Good]]>
            </sslSecurityFilters>
            <sslCiphers>
                <![CDATA[ECDH-ECDSA-AES128-GCM-SHA256,ECDH-RSA-AES128-
SHA256,ECDH-RSA-AES128-GCM-SHA256,DHE-RSA-AES128-GCM-SHA256,ECDHE-
ECDSA-AES128-SHA256,DH-RSA-AES128-SHA256,DH-RSA-AES256-SHA256,ECDH-
ECDSA-AES128-SHA256,DHE-RSA-AES256-SHA256,DH-RSA-AES128-GCM-
SHA256,ECDHE-RSA-AES128-GCM-SHA256,ECDHE-RSA-AES128-SHA256,ECDHE-
ECDSA-AES128-GCM-SHA256,DHE-RSA-AES128-SHA256]]>
            </sslCiphers>
            <blockingMode>false</blockingMode>
            <customPage>
                <id>1001</id>
                <uuid>0dba4434-1118-40e5-8768-23c5616053d5</uuid>
                <name>
                    <![CDATA[My Response]]>
                </name>
            </customPage>
            <securityPolicy>
                <id>30682</id>
                <uuid>6c56416a-66ff-4016-b16f-da2cec2e97f3</uuid>
                <name>
```

```
                <![CDATA[Standard Policy]]>
            </name>
        </securityPolicy>
        <httpProfile>
            <id>1001</id>
            <uuid>341bcf25-c9fa-45ff-ac63-728e38056443</uuid>
            <name>
                <![CDATA[Standard Protocol]]>
            </name>
        </httpProfile>
        <scanTrustEnabled>true</scanTrustEnabled>
        <scanTrustToken>
            <![CDATA[38770c30-7c79-4b75-a5ec-43d07493eca1]]>
        </scanTrustToken>
        <customRules>
            <CustomRule>
                <id>2001</id>
                <uuid>c64c3008-c1af-4969-8290-d0b1d8e9f27b</uuid>
                <name>
                    <![CDATA[shamzor]]>
                </name>
            </CustomRule>
        </customRules>
        <clusters>
            <Cluster>
                <id>24401</id>
                <uuid>48ae444d-e652-443f-8438-3a9182403b9f</uuid>
                <name>
                    <![CDATA[Cluster 1]]>
                </name>
            </Cluster>
        </clusters>
        <status>DOWN</status>
        <sslEnabled>true</sslEnabled>
        <sslStatus>OK</sslStatus>
        <deploymentStatus>PENDING_DEPLOY</deploymentStatus>
        <deployed>2017-06-02T16:10:06Z</deployed>
    </WebApp>
    <WebApp>
        <id>63098473</id>
        <uuid>6ecd55d8-5431-4114-ba11-90b020576f37</uuid>
        <name>
            <![CDATA[Site created by API]]>
        </name>
        <owner>
```

```
            <id>3988443</id>
            <username>john_doe</username>
            <firstname>John</firstname>
            <lastname>Doe</lastname>
        </owner>
        <created>2017-06-01T09:22:47Z</created>
        <createdBy>
            <id>3988443</id>
            <username>john_doe</username>
            <firstname>John</firstname>
            <lastname>Doe</lastname>
        </createdBy>
        <updated>2017-06-02T13:23:46Z</updated>
        <updatedBy>
            <id>3988443</id>
            <username>john_doe</username>
            <firstname>John</firstname>
            <lastname>Doe</lastname>
        </updatedBy>
        <url>https://site02.xfuentes-docker</url>
        <webServer>
            <id>1001</id>
            <uuid>315cc797-3c73-4721-ba42-263e7e7b6cbb</uuid>
            <name>
                <![CDATA[First Pool]]>
            </name>
        </webServer>
        <persistencyEnabled>false</persistencyEnabled>
        <healthcheck>
            <id>1001</id>
            <uuid>f479e6f5-57a1-4677-a8cb-272e2c69623a</uuid>
            <name>
                <![CDATA[Standard Healthcheck]]>
            </name>
        </healthcheck>
        <certificate>
            <id>1001</id>
            <uuid>5788c0eb-5bda-466f-bfb6-71a1f60856ff</uuid>
            <name>
                <![CDATA[Site02 Cert]]>
            </name>
        </certificate>
        <sslProtocols>
            <![CDATA[TLS12]]>
        </sslProtocols>
```

```
            <sslSecurityFilters>
                <![CDATA[Good]]>
            </sslSecurityFilters>
            <sslCiphers>
                <![CDATA[ECDH-ECDSA-AES128-GCM-SHA256,ECDH-RSA-AES128-
SHA256,ECDH-RSA-AES128-GCM-SHA256,DHE-RSA-AES128-GCM-SHA256,ECDHE-
ECDSA-AES128-SHA256,DH-RSA-AES128-SHA256,DH-RSA-AES256-SHA256,ECDH-
ECDSA-AES128-SHA256,DHE-RSA-AES256-SHA256,DH-RSA-AES128-GCM-
SHA256,ECDHE-RSA-AES128-GCM-SHA256,ECDHE-RSA-AES128-SHA256,ECDHE-
ECDSA-AES128-GCM-SHA256,DHE-RSA-AES128-SHA256]]>
            </sslCiphers>
            <blockingMode>true</blockingMode>
            <securityPolicy>
                <id>30682</id>
                <uuid>6c56416a-66ff-4016-b16f-da2cec2e97f3</uuid>
                <name>
                    <![CDATA[Standard Policy]]>
                </name>
            </securityPolicy>
            <httpProfile>
                <id>1001</id>
                <uuid>341bcf25-c9fa-45ff-ac63-728e38056443</uuid>
                <name>
                    <![CDATA[Standard Protocol]]>
                </name>
            </httpProfile>
            <scanTrustEnabled>false</scanTrustEnabled>
            <customRules>
                <CustomRule>
                    <id>2001</id>
                    <uuid>c64c3008-c1af-4969-8290-d0b1d8e9f27b</uuid>
                    <name>
                        <![CDATA[shamzor]]>
                    </name>
                </CustomRule>
            </customRules>
            <status>INACTIVE</status>
            <sslEnabled>true</sslEnabled>
            <sslStatus>OK</sslStatus>
            <deploymentStatus>PENDING_DEPLOY</deploymentStatus>
            <deployed>2017-06-01T16:05:54Z</deployed>
        </WebApp>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webapp.xsd

Delete web application

/qps/rest/2.0/delete/waf/webapp/<id>

[POST]

Delete a web application configuration in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Delete WAF Asset permissions, and web application licensed for WAF and within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the web application asset you want to delete.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/delete/waf/webapp/5739473"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/
xsd/2.0/waf/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>5739473</id>
    </WebApp>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webapp.xsd

Delete web applications (bulk)

/qps/rest/2.0/delete/waf/webapp

[POST]

Delete multiple web application assets in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Delete WAF Asset permissions, and web application licensed for WAF and within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search web applications.

Sample

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/delete/waf/webapp"

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

Request POST data

```
<?xml version="1.0" ?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">created by
API</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>63098473</id>
        </WebApp>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webapp.xsd

Web Servers API

Current web server count

/qps/rest/2.0/count/waf/webserver

[GET]

Returns the total number of web server pools for WAF in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission and within the user's scope.

Input Parameters

No input elements are available.

Sample

API request

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

Response

```
<?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
/waf/webserver.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>6</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webserver.xsd

Get details on a web server

/qps/rest/2.0/get/waf/webserver/<id>

[GET]

Returns details about a specific web server pool for WAF, within the user's scope. Want to find a web server ID to use as input? See Search web servers.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission and within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the web server pool ID of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/get/waf/webserver/2401"
```

Response

```
<?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
/waf/webserver.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebServer>
            <id>2401</id>
            <uuid>069446f8-dd5a-4c5e-8cbe-346c148582ba</uuid>
            <name>
                <![CDATA[WAF Web Server 6]]>
            </name>
            <description>
                <![CDATA[This is a server pool]]>
```

```
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-05-14T08:52:35Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-05-14T08:52:35Z</updated>
            <updatedBy>
                <id>3988443</id>
                 <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <loadBalancingAlgorithm>FIRST</loadBalancingAlgorithm>
            <addresses>
                <WebServerAddress>
                    <url>http://172.17.0.1:9081</url>
                    <weight>20</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://172.17.0.2:9081</url>
                    <weight>10</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://172.17.0.3:9081</url>
                    <weight>1</weight>
                </WebServerAddress>
            </addresses>
        </WebServer>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webserver.xsd

Search web servers

/qps/rest/2.0/search/waf/webserver

[POST]

Finds web server pools in the user's account matching the search criteria.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and assets must be 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. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Supported filter operators

Parameter	Description
id (Long)	Web server profile identifier on Qualys Cloud Platform.
uuid (UUID)	Web server profile identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the web server profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the web server profile.
loadBalancingAlgorithm (Text)	Choose the method to load balance traffic between the servers. Your choices are:

	roundrobin - Each server receives the connection in turns, according to their weights. Weights are dynamically adjusted for best performance. leastconn - The server with the lowest number of connections receives the connection. Use of this algorithm is recommended where very long sessions are expected, such as LDAP, SQL, TSE, etc. first - The first server with available connection slots receives the connection. The servers are chosen from the lowest numeric identifier to the highest, which defaults to the server's position in the farm. source - Only one designated server receives the connection, based on the source IP address. The source IP address is hashed and divided by the total weight of the running servers to designate which server will receive the request. This ensures that the same client IP address will always reach the same server as long as no server goes down or up.
addresses.url (Text)	You can add one server or multiple servers to load balance traffic between several original URLs. All URLs must use the same protocol and port. For each server provide the IP address or a Fully Qualified Domain Name.
addresses.weight (Integer)	For each server add the corresponding weight between 0 to 256. WAF uses weights to distribute the request load to various servers in the Web Server Pool. Default weight is 1.
owner.id (Long)	The user for Qualys Cloud Platform who owns this web server profile.
owner.username (Text)	The user name of the web server profile owner.

owner.firstname (Text)	The first name of the web server profile owner.
owner.lastname (Text)	The last name of the web server profile owner.
created (Date)	The date/time when the web server profile was created.
createdBy.id (Long)	The user ID who created the web server profile.
createdBy.username (Text)	The user name who created the web server profile.
createdBy.firstname (Text)	The first name of the user who created the web server profile.
createdBy.lastname (Text)	The last name of the user who created the web server profile.
updated (Date)	The date/time when the web server profile was last updated.
updatedBy.id (Long)	The user ID who last updated the web server profile.
updatedBy.username (Text)	The user name who last updated the web server profile.
updatedBy.firstname (Text)	The first name of the user who updated the web server profile.
updatedBy.lastname (Text)	The last name of the user who updated the web server profile.
webApps.webApp.id (Long)	The ID of the Web Application this web server profile is associated with.
webApps.webApp.uuid (UUID)	The UUID of the Web Application this web server profile is associated with.
webApps.webApp.name	The name of the Web Application this web

(Text)	server profile is associated with.
tags.tag.id (Long)	The ID of a tag associated with the web server profile.
tags.tag.name (Text)	The name, defined by a user, of a tag associated with the web server profile.

Sample

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/webserver"

Note: "file.xml" contains the request POST data.
The request POST data is optional. If you leave it empty all web
server pools in the user’s scope are returned.
```

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <filters>
        <Criteria field="addresses.url"
operator="CONTAINS">172.17.0</Criteria>
        <Criteria field="addresses.weight"
operator="GREATER">1</Criteria>
    </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/webserver.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <WebServer>
```

```
            <id>2401</id>
            <uuid>069446f8-dd5a-4c5e-8cbe-346c148582ba</uuid>
            <name>
                <![CDATA[WAF Web Server 6]]>
            </name>
            <description>
                <![CDATA[This is a server pool]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-05-14T08:52:35Z</created>
            <createdBy>
                <id>3988443</id>
                 <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-05-14T08:52:35Z</updated>
            <updatedBy>
                <id>3988443</id>
               <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <loadBalancingAlgorithm>FIRST</loadBalancingAlgorithm>
            <addresses>
                <WebServerAddress>
                    <url>http://172.17.0.1:9081</url>
                    <weight>20</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://172.17.0.2:9081</url>
                    <weight>10</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://172.17.0.3:9081</url>
                    <weight>1</weight>
                </WebServerAddress>
            </addresses>
        </WebServer>
    </data>
```

```
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webserver.xsd

Create web server

/qps/rest/2.0/create/waf/webserver

[POST]

Create a web server pool which you can assign to a web application.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Create WAF Asset permissions.

Input Parameters

Allowed input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	(Required) The name of the web server profile as defined by a user. This is unique in subscription. Valid action: Update
loadBalancingAlgorithm (Text)	(Required) Choose the method to load balance traffic between the servers. Your choices are: **roundrobin** - Each server receives the connection in turns, according to their weights. Weights are dynamically adjusted for best performance. **leastconn** - The server with the lowest number of connections receives the connection. Use of this algorithm is recommended where very long sessions

	are expected, such as LDAP, SQL, TSE, etc. **first** - The first server with available connection slots receives the connection. The servers are chosen from the lowest numeric identifier to the highest, which defaults to the server's position in the farm. **source** - Only one designated server receives the connection, based on the source IP address. The source IP address is hashed and divided by the total weight of the running servers to designate which server will receive the request. This ensures that the same client IP address will always reach the same server as long as no server goes down or up.
addresses.WebServerAddress	(Required) At least one backend web server address. Each of them should contain an url and a weight element. weight is 1 by default. Each addresses.WebServerAddress element should contain a url and a weight.
description (Text)	The description of the web server profile.
tags	List of tags associated with the web server profile. Valid action: Update
tags.tag.id (Long)	The ID of a tag associated with the web server profile.
tags.tag.name (Text)	The name, defined by a user, of a tag associated with the web server profile.

Sample

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/webserver"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <WebServer>
            <name>WAF Web Server 6</name>
            <description>This is a server pool</description>
            <loadBalancingAlgorithm>FIRST</loadBalancingAlgorithm>
            <addresses>
                <WebServerAddress>
                    <url>http://172.17.0.1:9081</url>
                    <weight>20</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://172.17.0.2:9081</url>
                    <weight>10</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://172.17.0.3:9081</url>
                </WebServerAddress>
            </addresses>
        </WebServer>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/webserver.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebServer>
            <id>2401</id>
            <uuid>069446f8-dd5a-4c5e-8cbe-346c148582ba</uuid>
```

```
            <name>
                <![CDATA[WAF Web Server 6]]>
            </name>
            <description>
                <![CDATA[This is a server pool]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-05-14T08:52:35Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-05-14T08:52:35Z</updated>
            <updatedBy>
                <id>3988443</id>
                 <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <loadBalancingAlgorithm>FIRST</loadBalancingAlgorithm>
            <addresses>
                <WebServerAddress>
                    <url>http://172.17.0.1:9081</url>
                    <weight>20</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://172.17.0.2:9081</url>
                    <weight>10</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://172.17.0.3:9081</url>
                    <weight>1</weight>
                </WebServerAddress>
            </addresses>
        </WebServer>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webserver.xsd

Update web server

/qps/rest/2.0/update/waf/webserver/<id>

[POST]

Update a web server pool in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Update WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the web server pool you want to update.

Optional input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	The name of the web server profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the web server profile.
loadBalancingAlgorithm (Text)	Choose the method to load balance traffic between the servers. Your choices are: **roundrobin** - Each server receives the connection in turns, according to their weights. Weights are dynamically adjusted

	for best performance. **leastconn** - The server with the lowest number of connections receives the connection. Use of this algorithm is recommended where very long sessions are expected, such as LDAP, SQL, TSE, etc. **first** - The first server with available connection slots receives the connection. The servers are chosen from the lowest numeric identifier to the highest, which defaults to the server's position in the farm. **source** - Only one designated server receives the connection, based on the source IP address. The source IP address is hashed and divided by the total weight of the running servers to designate which server will receive the request. This ensures that the same client IP address will always reach the same server as long as no server goes down or up.
addresses.WebServerAddress	At least one backend web server address. Each of them should contain an url and a weight element. weight is 1 by default. Each addresses.WebServerAddress element should contain a url and a weight.
tags	List of tags associated with the web server profile. Valid action: Update

Sample

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/webserver/2401"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <WebServer>
            <name>WAF Web Server 11</name>
            <description>New server pool</description>
       <loadBalancingAlgorithm>ROUNDROBIN</loadBalancingAlgorithm>
            <addresses>
                <WebServerAddress>
                    <url>http://55.17.0.1:9081</url>
                    <weight>5</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://56.17.0.2:9081</url>
                    <weight>4</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://57.17.0.3:9081</url>
                    <weight>3</weight>
                </WebServerAddress>
            </addresses>
        </WebServer>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/webserver.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebServer>
            <id>2401</id>
            <uuid>069446f8-dd5a-4c5e-8cbe-346c148582ba</uuid>
```

```
            <name>
                <![CDATA[WAF Web Server 11]]>
            </name>
            <description>
                <![CDATA[New server pool]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-05-14T08:52:35Z</created>
            <createdBy>
                <id>3988443</id>
                 <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-05-14T09:21:16Z</updated>
            <updatedBy>
                <id>3988443</id>
                 <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
       <loadBalancingAlgorithm>ROUNDROBIN</loadBalancingAlgorithm>
            <addresses>
                <WebServerAddress>
                    <url>http://56.17.0.2:9081</url>
                    <weight>4</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://55.17.0.1:9081</url>
                    <weight>5</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://57.17.0.3:9081</url>
                    <weight>3</weight>
                </WebServerAddress>
            </addresses>
        </WebServer>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webserver.xsd

Update web servers (bulk)

/qps/rest/2.0/update/waf/webserver

[POST]

Update multiple web server pool in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Update WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search web servers.

Allowed input elements for bulk update are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Supported filter operators

Parameter	Description
name (Text)	The name of the web server profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the web server profile.
loadBalancingAlgorithm (Text)	Choose the method to load balance traffic between the servers. Your choices are: **roundrobin** - Each server receives the connection in turns, according to their

	weights. Weights are dynamically adjusted for best performance. **leastconn** - The server with the lowest number of connections receives the connection. Use of this algorithm is recommended where very long sessions are expected, such as LDAP, SQL, TSE, etc. **first** - The first server with available connection slots receives the connection. The servers are chosen from the lowest numeric identifier to the highest, which defaults to the server's position in the farm. **source** - Only one designated server receives the connection, based on the source IP address. The source IP address is hashed and divided by the total weight of the running servers to designate which server will receive the request. This ensures that the same client IP address will always reach the same server as long as no server goes down or up.
addresses.WebServerAddress	At least one backend web server address. Each of them should contain an url and a weight element. weight is 1 by default. Each addresses.WebServerAddress element should contain a url and a weight.
tags	List of tags associated with the web server profile. Valid action: Update

Sample

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/webserver"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="loadBalancingAlgorithm" operator="NOT
EQUALS">ROUNDROBIN</Criteria>
  </filters>
  <data>
    <WebServer>
<loadBalancingAlgorithm>ROUNDROBIN</loadBalancingAlgorithm>
    </WebServer>
  </data>
</ServiceRequest>
```

Response

```
<?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
/waf/webserver.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebServer>
            <id>2401</id>
            <uuid>069446f8-dd5a-4c5e-8cbe-346c148582ba</uuid>
            <name>
                <![CDATA[WAF Web Server 11]]>
            </name>
            <description>
                <![CDATA[New server pool]]>
            </description>
            <owner>
                <id>3988443</id>
                   <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
```

```
            <created>2017-05-14T08:52:35Z</created>
            <createdBy>
                <id>3988443</id>
                    <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-05-14T09:25:31Z</updated>
            <updatedBy>
                <id>3988443</id>
                   <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <loadBalancingAlgorithm>ROUNDROBIN</loadBalancingAlgorithm
>
            <addresses>
                <WebServerAddress>
                    <url>http://56.17.0.2:9081</url>
                    <weight>4</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://55.17.0.1:9081</url>
                    <weight>5</weight>
                </WebServerAddress>
                <WebServerAddress>
                    <url>http://57.17.0.3:9081</url>
                    <weight>3</weight>
                </WebServerAddress>
            </addresses>
        </WebServer>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webserver.xsd

Delete web server

/qps/rest/2.0/delete/waf/webserver/<id>

[POST]

Delete a web server pool in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Delete WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the web server pool you want to delete.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/delete/waf/webserver/1201"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://lqualysapi.qualys.com/qps/xsd/2.
0/waf/webserver.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebServer>
            <id>1201</id>
        </WebServer>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webserver.xsd

Delete web servers (bulk)

/qps/rest/2.0/delete/waf/webserver

[POST]

Delete multiple web server pools in user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Delete WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search web servers.

Sample

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/delete/waf/webserver"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
<filters>
<Criteria field="name" operator="CONTAINS">My Web Server</Criteria>
</filters>
</ServiceRequest>
```

Response

```
<?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
/waf/webserver.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <WebServer>
            <id>1401</id>
        </WebServer>
        <WebServer>
            <id>1601</id>
        </WebServer>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/webserver.xsd

Healthchecks API

Current healthcheck count

/qps/rest/2.0/count/waf/healthcheck

[GET]

Returns the total number of healthchecks for WAF in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission and asset must be within the user's scope.

Input Parameters

No input elements are available.

Sample

API request

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

Response

```
<?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
/waf/healthcheck.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>3</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/healthcheck.xsd

Get details on a healthcheck

/qps/rest/2.0/get/waf/healthcheck/<id>

[GET]

Returns details about a specific healthcheck for WAF, within the user's scope. Want to find a healthcheck ID to use as input? See Search healthchecks.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission and asset must be within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the healthcheck ID of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/get/waf/healthcheck/1401"
```

Response

```
<?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
/waf/healthcheck.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Healthcheck>
            <id>1401</id>
            <uuid>b491ed9c-d9fd-461d-81eb-e8ee251289c7</uuid>
            <name>
                <![CDATA[My Healthcheck]]>
            </name>
            <description>
```

```
                <![CDATA[This is a healthcheck]]>
            </description>
            <owner>
                <id>3988443</id>
                 <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>            </owner>
            <created>2018-11-11T08:29:42Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2018-11-11T08:29:42Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <system>false</system>
            <method>GET</method>
            <path>
                <![CDATA[/path]]>
            </path>
            <expectedResponseCode>200</expectedResponseCode>
            <intervalUp>5</intervalUp>
            <intervalDown>6</intervalDown>
            <intervalFlapping>7</intervalFlapping>
            <nbSuccessesUp>8</nbSuccessesUp>
            <nbFailuresDown>9</nbFailuresDown>
            <timeout>10</timeout>
        </Healthcheck>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/healthcheck.xsd

Search healthchecks

/qps/rest/2.0/search/waf/healthcheck

[POST]

Finds healthchecks in the user's account matching the search criteria.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and asset must be 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. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Supported filter operators

Parameter	Description
id (Long)	Healthcheck profile identifier on Qualys Cloud Platform.
uuid (UUID)	Healthcheck profile identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the Healthcheck profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the Healthcheck profile.
method (Text)	The method to use in the healthcheck periodic request

	(GET or HEAD).
path (Text)	The path to use in the healthcheck periodic request. Can be empty but have to be specified.
expectedResponseCode (Long)	Expected web server response code to consider the web server as available.
intervalUp (Long)	Number of healthcheck requests per second when the web server status is up.
intervalDown (Long)	Number of healthcheck requests per second when the web server status is down.
intervalFlapping (Long)	Number of healthcheck requests per second when the web server status is flapping (not stable).
nbSuccessesUp (Long)	The amount of successes before considering the web server as up.
nbFailuresDown (Long)	The amount of failures before considering the web server as down.
timeout (Long)	The healthcheck request timeout in seconds.
owner (Text)	The user for Qualys Cloud Platform who owns this healthcheck profile.
owner.id (Long)	The user ID of the healthcheck profile owner.
owner.username (Text)	The user name of the healthcheck profile owner.
owner.firstname (Text)	The first name of the healthcheck profile owner.
owner.lastname (Text)	The last name of the healthcheck profile owner.
created (Date)	The date/time when the healthcheck profile was created.
createdBy.id (Long)	The user ID who created the healthcheck profile.

createdBy.username (Text)	The user name who created the healthcheck profile.
createdBy.firstname (Text)	The first name of the user who created the healthcheck profile.
createdBy.lastname (Text)	The last name of the user who created the healthcheck profile.
updated (Date)	The date/time when the healthcheck profile was last updated.
updatedBy.id (Long)	The user ID who last updated the healthcheck profile.
updatedBy.username (Text)	The user name who last updated the healthcheck profile.
updatedBy.firstname (Text)	The first name of the user who updated the healthcheck profile.
updatedBy.lastname (Text)	The last name of the user who updated the healthcheck profile.
webApps.webApp.id (Long)	The ID of the Web Application this healthcheck profile is associated with.
webApps.webApp.uuid (UUID)	The UUID of the Web Application this healthcheck profile is associated with.
webApps.webApp.name (Text)	The name of the Web Application this healthcheck profile is associated with.
tags.tag.id (Long)	The ID of a tag associated with the healthcheck profile.
tags.tag.name (Text)	The name, defined by a user, of a tag associated with the healthcheck profile.

Sample

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/healthcheck"

Note: "file.xml" contains the request POST data.
The request POST data is optional. If you leave it empty all
healthchecks in the user's scope are returned.
```

Request POST data

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

Response

```
<?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
/waf/healthcheck.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <Healthcheck>
            <id>10801</id>
            <uuid>0e77cf97-b33a-4105-b273-72d49217b565</uuid>
            <name>
                <![CDATA[Standard Healthcheck]]>
            </name>
            <owner>
                <id>2501190</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-03-01T22:22:28Z</created>
            <createdBy>
```

```
                <id>2501190</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-03-01T22:22:28Z</updated>
            <system>true</system>
            <method>GET</method>
            <path>
                <![CDATA[/]]>
            </path>
            <expectedResponseCode>200</expectedResponseCode>
            <intervalUp>15</intervalUp>
            <intervalDown>5</intervalDown>
            <intervalFlapping>10</intervalFlapping>
            <nbSuccessesUp>3</nbSuccessesUp>
            <nbFailuresDown>3</nbFailuresDown>
            <timeout>15</timeout>
            <webApps/>
        </Healthcheck>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/healthcheck.xsd

Create healthcheck

/qps/rest/2.0/create/waf/healthcheck

[POST]

Creates a healthcheck which you can assign to a web application.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Create WAF Asset permissions.

Input Parameters

Allowed input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	(Required) The name of the Healthcheck profile as defined by a user. This is unique in subscription. Valid action: Update
method (Text)	(Required) The method to use in the healthcheck periodic request (GET or HEAD).
path (Text)	(Required) The path to use in the healthcheck periodic request. Can be empty but have to be specified.
expectedResponseCode (Long)	(Required) Expected web server response code to consider the web server as available.
intervalUp (Long)	(Required) Number of healthcheck requests per

	second when the web server status is up.
intervalDown (Long)	(Required) Number of healthcheck requests per second when the web server status is down.
intervalFlapping (Long)	(Required) Number of healthcheck requests per second when the web server status is flapping (not stable).
nbSuccessesUp (Long)	(Required)The amount of successes before considering the web server as up.
nbFailuresDown (Long)	(Required) The amount of failures before considering the web server as down.
timeout (Long)	(Required) The healthcheck request timeout in seconds.
description (Text)	The description of the Healthcheck profile.
tags	List of tags associated with the healthcheck profile. Valid action: Update
tags.tag.id (Long)	The ID of a tag associated with the healthcheck profile.
tags.tag.name (Text)	The name, defined by a user, of a tag associated with the healthcheck profile.

Sample

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/healthcheck"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <Healthcheck>
            <name>My Healthcheck</name>
            <description>desc</description>
            <method>GET</method>
            <path>/path</path>
            <expectedResponseCode>200</expectedResponseCode>
            <intervalUp>5</intervalUp>
            <intervalDown>6</intervalDown>
            <intervalFlapping>7</intervalFlapping>
            <nbSuccessesUp>8</nbSuccessesUp>
            <nbFailuresDown>9</nbFailuresDown>
            <timeout>10</timeout>
        </Healthcheck>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/healthcheck.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Healthcheck>
            <id>1601</id>
            <uuid>1b743e15-0756-4ac9-870a-b71dc031c2d4</uuid>
            <name>
                <![CDATA[My Healthcheck]]>
            </name>
            <description>
                <![CDATA[This is a healthcheck]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-04-11T13:54:37Z</created>
```

```
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-04-11T13:54:37Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <system>false</system>
            <method>GET</method>
            <path>
                <![CDATA[/path]]>
            </path>
            <expectedResponseCode>200</expectedResponseCode>
            <intervalUp>5</intervalUp>
            <intervalDown>6</intervalDown>
            <intervalFlapping>7</intervalFlapping>
            <nbSuccessesUp>8</nbSuccessesUp>
            <nbFailuresDown>9</nbFailuresDown>
            <timeout>10</timeout>
        </Healthcheck>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/healthcheck.xsd

Update healthcheck

/qps/rest/2.0/update/waf/healthcheck/<id>

[POST]

Update a healthcheck in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Update WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the healthcheck you want to update.

Optional input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	The name of the Healthcheck profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the Healthcheck profile.
method (Text)	The method to use in the healthcheck periodic request (GET or HEAD).
path (Text)	The path to use in the healthcheck periodic request. Can be empty but have to be specified.

expectedResponseCode (Long)	Expected web server response code to consider the web server as available.
intervalUp (Long)	Number of healthcheck requests per second when the web server status is up.
intervalDown (Long)	Number of healthcheck requests per second when the web server status is down.
intervalFlapping (Long)	Number of healthcheck requests per second when the web server status is flapping (not stable).
nbSuccessesUp (Long)	The amount of successes before considering the web server as up.
nbFailuresDown (Long)	The amount of failures before considering the web server as down.
timeout (Long)	The healthcheck request timeout in seconds.
tags	List of tags associated with the healthcheck profile. Valid action: Update

Sample

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/healthcheck/1602
"

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

Request POST data

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

```
    <data>
        <Healthcheck>
            <description>New healthcheck desc</description>
            <intervalFlapping>77</intervalFlapping>
        </Healthcheck>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/healthcheck.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Healthcheck>
            <id>1602</id>
            <uuid>f032764e-1de3-49e7-9c22-a9f070a909ca</uuid>
            <name>
                <![CDATA[My Healthcheck 2]]>
            </name>
            <description>
                <![CDATA[New healthcheck desc]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-04-11T13:57:41Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-04-11T13:58:58Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
```

```
            </updatedBy>
            <system>false</system>
            <method>GET</method>
            <path>
                <![CDATA[/path]]>
            </path>
            <expectedResponseCode>200</expectedResponseCode>
            <intervalUp>5</intervalUp>
            <intervalDown>6</intervalDown>
            <intervalFlapping>77</intervalFlapping>
            <nbSuccessesUp>8</nbSuccessesUp>
            <nbFailuresDown>9</nbFailuresDown>
            <timeout>10</timeout>
        </Healthcheck>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/healthcheck.xsd

Update healthchecks (bulk)

/qps/rest/2.0/update/waf/healthcheck

[POST]

Update multiple healthchecks in the user's account. You can update all fields except tagID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Update WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search healthchecks.

Allowed input elements for bulk update are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Supported filter operators

Parameter	Description
name (Text)	The name of the Healthcheck profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the Healthcheck profile.
method (Text)	The method to use in the healthcheck periodic request (GET or HEAD).
path (Text)	The path to use in the healthcheck periodic

	request. Can be empty but have to be specified.
expectedResponseCode (Long)	Expected web server response code to consider the web server as available.
intervalUp (Long)	Number of healthcheck requests per second when the web server status is up.
intervalDown (Long)	Number of healthcheck requests per second when the web server status is down.
intervalFlapping (Long)	Number of healthcheck requests per second when the web server status is flapping (not stable).
nbSuccessesUp (Long)	The amount of successes before considering the web server as up.
nbFailuresDown (Long)	The amount of failures before considering the web server as down.
timeout (Long)	The healthcheck request timeout in seconds.
tags	List of tags associated with the healthcheck profile. Valid action: Update

Sample

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/healthcheck"

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

Request POST data

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

```
<ServiceRequest>
  <filters>
    <Criteria field="nbSuccessesUp" operator="GREATER">5</Criteria>
  </filters>
  <data>
    <Healthcheck>
        <nbFailuresDown>55</nbFailuresDown>
    </Healthcheck>
  </data>
</ServiceRequest>
```

Response

```
<?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
/waf/healthcheck.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <Healthcheck>
            <id>1601</id>
            <uuid>1b743e15-0756-4ac9-870a-b71dc031c2d4</uuid>
            <name>
                <![CDATA[My Healthcheck]]>
            </name>
            <description>
                <![CDATA[This is a healthcheck]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-04-11T13:54:37Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-04-11T14:09:13Z</updated>
            <updatedBy>
                <id>3988443</id>
```

```
            <username>john_doe</username>
            <firstname>John</firstname>
            <lastname>Doe</lastname>
        </updatedBy>
        <system>false</system>
        <method>GET</method>
        <path>
            <![CDATA[/path]]>
        </path>
        <expectedResponseCode>200</expectedResponseCode>
        <intervalUp>5</intervalUp>
        <intervalDown>6</intervalDown>
        <intervalFlapping>7</intervalFlapping>
        <nbSuccessesUp>99</nbSuccessesUp>
        <nbFailuresDown>55</nbFailuresDown>
        <timeout>10</timeout>
    </Healthcheck>
    <Healthcheck>
        <id>1602</id>
        <uuid>f032764e-1de3-49e7-9c22-a9f070a909ca</uuid>
        <name>
            <![CDATA[My Healthcheck 2]]>
        </name>
        <description>
            <![CDATA[my desc]]>
        </description>
        <owner>
            <id>3988443</id>
            <username>john_doe</username>
            <firstname>John</firstname>
            <lastname>Doe</lastname>
        </owner>
        <created>2017-04-11T13:57:41Z</created>
        <createdBy>
            <id>3988443</id>
            <username>john_doe</username>
            <firstname>John</firstname>
            <lastname>Doe</lastname>
        </createdBy>
        <updated>2017-04-11T14:09:17Z</updated>
        <updatedBy>
            <id>3988443</id>
            <username>john_doe</username>
            <firstname>John</firstname>
            <lastname>Doe</lastname>
```

```
            </updatedBy>
            <system>false</system>
            <method>GET</method>
            <path>
                <![CDATA[/path]]>
            </path>
            <expectedResponseCode>200</expectedResponseCode>
            <intervalUp>5</intervalUp>
            <intervalDown>6</intervalDown>
            <intervalFlapping>77</intervalFlapping>
            <nbSuccessesUp>99</nbSuccessesUp>
            <nbFailuresDown>55</nbFailuresDown>
            <timeout>10</timeout>
        </Healthcheck>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/healthcheck.xsd

Delete healthcheck

/qps/rest/2.0/delete/waf/healthcheck/<id>

[POST]

Delete a healthcheck in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Delete WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the healthcheck you want to delete.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/delete/waf/healthcheck/1402
"
```

Response

```
<?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
/waf/healthcheck.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Healthcheck>
            <id>1402</id>
        </Healthcheck>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/healthcheck.xsd

Delete healthchecks (bulk)

/qps/rest/2.0/delete/waf/healthcheck

[POST]

Delete multiple healthchecks in user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Delete WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search healthchecks.

Sample

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/delete/waf/healthcheck"

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

Request POST data

```
<?xml version="1.0" ?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">My
Healthcheck</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/healthcheck.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Healthcheck>
            <id>1401</id>
        </Healthcheck>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/healthcheck.xsd

SSL Certificates API

Current certificate count

/qps/rest/2.0/count/waf/certificate

[GET]

Returns the total number of SSL certificates for WAF in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission and asset must be within the user's scope.

Input Parameters

No input elements are available.

Sample

API request

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

Response

```
<?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
/waf/certificate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>3</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/certificate.xsd

Get details on a certificate

/qps/rest/2.0/get/waf/certificate/<id>

[GET]

Returns details about a specific SSL certificate for WAF, within the user's scope. Want to find an SSL certificate ID to use as input? See Search certificates.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission and asset must be within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the SSL certificate ID of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/get/waf/certificate/8"
```

Response

```
<?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
/waf/certificate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Certificate>
            <id>8</id>
            <uuid>a89fe014-d1c9-446d-b38f-8f7726158595</uuid>
            <name>
                <![CDATA[Site03 PKCS12 Certificate]]>
            </name>
```

```
            <description>
                <![CDATA[This is a certificate]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-04-04T07:50:12Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-04-04T07:50:12Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <certificateMetadata>
                <![CDATA[{"fileName":"uploaded-
cert.pfx","commonName":"site03.xfuentes-
docker","issuer":"EMAILADDRESS=xfuentes@qualys.com, CN=Intermediate
CA, OU=Engineering, O=Qualys, ST=France,
C=FR","dateStart":1490021391000,"dateEnd":1522421391000,"subject":"EMA
ILADDRESS=xfuentes@qualys.com, CN=site03.xfuentes-docker,
OU=Engineering, O=Qualys, L=Carcassonne, ST=France,
C=FR","sigAlgo":"SHA256WithRSAEncryption","sn":"4101","version":3,"isE
xpired":false,"isYetValid":true,"isSelfSigned":false,"publicKey":"UlNB
IFB1YmxpYyBLZXkKICAgICAgICAgICAgbW9kdWx1czogZTFkYWFiMmQ0NjNkOWQxZTIyNG
YwZmU1ZDUyYzY2NDQ1NWE4NzBmY2RhNWQyMzRjM2U3NWYxNGNiYjUwM2Q3ODVmZWY0NjYz
Y2E3ePassphrase":true,"content":"LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktL
S0t10TkRUUG8vTworeVp3a2RCZjJ5NHhQZ0FSUG5rbkRLRFlCNnJwQUpyOFY1anY55Vnow
VjlWR0gwa2VxK3NQQUUvbW1OSG1hWE9Tdm9YekRtbllIZnM5azFNOAotLS0tLUVORCBSU0
EgUFJJVkFURSBLRVktLS0tLQo=","fileName":"uploaded-
cert.pfx","passphrase":"ssl","customerToken":"testtoken","publicKey":"
UlNBIFB1YmxpYyBLZXkKICAgICAgICAgICAgbW9kdWx1czogZTFkYWFiMmQ0NjNkOWQxZT
IyNGYwZmU1ZMNjhZmNkOGFjODU3Mjg2NjJiNWJjY2JhM2Y2NzkzMTdlY2U1ZDRhNzFlOGM
3Y2MxNGZlYTg5ZWQzZDA4MGFiMGFiOGYwYzU1ZmNjYjkyNGY0NzMyOTBkNjVlMzUKICAgI
HB1YmxpYyBleHBvbmVudDogMTAwMDEK"}}...]]>
            </certificateMetadata>
```

```
        </Certificate>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/certificate.xsd

Search certificates

/qps/rest/2.0/search/waf/certificate

[POST]

Finds SSL Certificates in the user's account matching the search criteria.

Permissions 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. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Supported filter operators

Parameter	Description
id (Long)	Certificate profile identifier on Qualys Cloud Platform.
uuid (UUID)	Certificate profile identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the certificate profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the certificate profile.
owner.id (Long)	The user ID of the certificate profile owner.

owner.username (Text)	The user name of the certificate profile owner.
owner.firstname (Text)	The first name of the certificate profile owner.
owner.lastname (Text)	The last name of the certificate profile owner.
created (Date)	The date/time when the certificate profile was created.
createdBy.id (Long)	The user ID who created the certificate profile.
createdBy.username (Text)	The user name who created the certificate profile.
createdBy.firstname (Text)	The first name of the user who created the certificate profile.
createdBy.lastname (Text)	The last name of the user who created the certificate profile.
updated (Date)	The date/time when the certificate profile was last updated.
updatedBy.id (Long)	The user ID who last updated the certificate profile.
updatedBy.username (Text)	The user name who last updated the certificate profile.
updatedBy.firstname (Text)	The first name of the user who updated the certificate profile.
updatedBy.lastname (Text)	The last name of the user who updated the certificate profile.
webApps.webApp.id (Long)	The ID of the Web Application this certificate profile is associated with.
webApps.webApp.uuid (UUID)	The UUID of the Web Application this certificate profile is associated with.

webApps.webApp.name (Text)	The name of the Web Application this certificate profile is associated with.
tags.tag.id (Long)	The ID of a tag associated with the certificate profile.
tags.tag.name (Text)	The name, defined by a user, of a tag associated with the certificate profile.

Sample

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/certificate"

Note: "file.xml" contains the request POST data.
The request POST data is optional. If you leave it empty all
certificates in the user’s scope are returned.
```

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
   <filters>
      <Criteria field="name" operator="CONTAINS">SUB</Criteria>
   </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/certificate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <Certificate>
            <id>36201</id>
            <uuid>52d35d06-b6bf-4365-bbee-81fea7a81115</uuid>
```

```
            <name>
                <![CDATA[SubUserSSL]]>
            </name>
            <owner>
                <id>361390</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-02-22T00:22:32Z</created>
            <createdBy>
                <id>361390</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-02-23T00:31:51Z</updated>
            <updatedBy>
                <id>361390</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <tags>
                <Tag>
                    <id>7531232</id>
                    <name>
                        <![CDATA[Unassigned Business Unit]]>
                    </name>
                </Tag>
            </tags>
            <certificateMetadata>
                <![CDATA[{"fileName":"leaf.pfx","commonName":"*.eng.qu
alys.com","issuer":"EMAILADDRESS=vmatosyan@qualys.com, CN=level1, OU=R
& D, O=Qualys Inc., L=Redwood City, ST=California,
C=US","dateStart":1452544631000,"dateEnd":1530304631000,"subject":"EMA
ILADDRESS=vmatosyan@qualys.com, CN=*.eng.qualys.com, OU=R & D,
O=Qualys Inc., L=Redwood City, ST=California,
C=US","sigAlgo":"MD5WithRSAEncryption","sn":"1","version":3,"isExpired
":false,"isYetValid":true,"isSelfSigned":false,"publicKey":"UlNBIFB1Ym
xpYyBLZXkKICAgICAgICAgICAgbW9kdWx1czogZGQ1ZmUwYTNhNTE0NzA4M2FjMDk0NGQz
ZDI4Mjk4dDogMTAwMDEK","subjectAltNames":["eng.qualys.com"],"subjectOrg
anization":"Qualys
Inc.","subjectEmail":"vmatosyan@qualys.com","issuerOrganization":"Qual
ys
```

```
Inc.","publicKeySize":768,"privateKey":{"usePassphrase":true,"content"
:"LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpQcm9jLVR5cGU6IDQsRU5DUllQ
VEVECkRFSy1JbmZvOiBERVMtRURFMy1DQkMsNmRmNmVkYjgyMmMwMGM3MwoKZnUzcUxLVE
NBNEsAwYjliMjM2ZDYWQ4MjkKICAgIHB1YmxpYyBleHBvbmVudDogMTAwMDEK"}}...]]>
            </certificateMetadata>
        </Certificate>
    </data>
</ServiceResponse>
```


XSD

<platform API server>/qps/xsd/2.0/waf/certificate.xsd

Create certificate

/qps/rest/2.0/create/waf/certificate

[POST]

Creates an SSL Certificate which you can assign to a web application. A certificate profile can be created in two ways:

- You can create a certificate profile using a pfx file. In this case you must provide a pkcs12 element containing a cdata with the base64 encoded content of the pfx file and a passphrase used to decrypt the file. See Using a pfx file

- You can create a certificate profile using a certificate encoded as PEM along with it's private key file as PEM. In this case you must provide a certificate and privateKey element containing a cdata with the base64 encoded content of the PEM files and a passphrase used to decrypt the private key. See Using a PEM file

Input Parameters

Allowed input elements are listed below. The associated data type for each element appears in parentheses.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Create WAF Asset permissions.

Input Parameters

Allowed input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	(Required) The name of the certificate profile as

	defined by a user. This is unique in subscription. Valid action: Update
passphrase (Text)	(Required) Used to decrypt the provided PFX or PEM private key.
token (Text)	(Required) The privateKey of the certificate will be encrypted using this token. This token have to be specified on the WAF appliance where the certificate is to be installed to be able to decrypt it. Use the WAF appliance CLI to provided the token as the waf_ssl_passphrase.
description (Text)	The description of the certificate profile.
pkcs12 (Text)	base64 encoded PFX file content (containing certificate and private key). The passphrase is required to decrypt the file.
certificate (Text)	base64 encoded PEM file content of the certificate (requires privateKey attribute).
privateKey (Text)	base64 encoded PEM file content of the private key. The passphrase is required if the key file is encrypted.
chain (Text)	base64 encoded PEM file content of the certificate authority chain certificates.
tags	List of tags associated with the certificate profile. Valid action: Update
tags.tag.id (Long)	The ID of a tag associated with the certificate profile.
tags.tag.name (Text)	The name, defined by a user, of a tag associated with the certificate profile.

Sample

Using a pfx file

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/certificate"

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

Request POST data

```
<ServiceRequest>
  <data>
    <Certificate>
      <name>Site01 Certificate</name>
      <pkcs12>
    <![CDATA[MIIMQQIBAzCCDAcGCSqGSIb3DQEHAaCCC/gEggv0MIIL8DCCBqcGCSqGS
Ib3DQEHBqCCBpgwggaUAgEAMIIGjQYJKoZI+EfSAdX3p0yHbkfMDEwITAJBgUrDgMCGgUA
BBQYWn1qCxhOJ9oxXNMso//j3aadWQQImOh0187j8kMCAggA...]]>
      </pkcs12>
      <passphrase><![CDATA[ssl]]></passphrase>
    </Certificate>
  </data>
</ServiceRequest>
```

Response

```
<?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
/waf/certificate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Certificate>
            <id>1</id>
            <uuid>15322a6c-e936-483a-8f76-ad0947be9bed</uuid>
            <name>
                <![CDATA[Site01 Certificate]]>
            </name>
            <owner>
                <id>3988443</id>
```

```
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-04-03T09:01:03Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-04-03T09:01:03Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <token>
                <![CDATA[3c53911c79d6c6b10878768128548a10bfc9c3785f00a
dd9760625d08ea00656d26682933cd07747f65caaac3390bbc6a1eee60f2dc2221317a
01a9ee26f3c7c69290baa9dd6939a96ce9d4055aaf9f054a26a47d32...]]>
            </token>
            <certificateMetadata>
                <![CDATA[{"fileName":"uploaded-
cert.pfx","commonName":"site03.xfuentes-
docker","issuer":"EMAILADDRESS=xfuentes@qualys.com, CN=Intermediate
CA, OU=Engineering, O=Qualys, ST=France,
C=FR","dateStart":1490021391000,"dateEnd":1522421391000,"subject":"EMA
ILADDRESS=xfuentes@qualys.com, CN=site03.xfuentes-docker,
OU=Engineering, O=Qualys, L=Carcassonne, ST=France,
C=FR","sigAlgo":"SHA256WithRSAEncryption","sn":"4101","version":3,"isE
xpired":false,"isYetValid":true,"isSelfSigned":false,"publicKey":"UlNB
IFB1YmxpYyBLZXkKICAgICAgICAgICAgbW9kdWx1czogZTFkYWFiMmQ0NjNkOWQxZTIyNS
1sZ5K2NpZHFTGhkVkxN21ZDRhNzFlOGM3Y2MxNGZlYTg5ZWQzZDA4MGFiMGFiOGYwYzU1Z
mNjYjkyNGY0lMzUKICAgIHB1BudDogMTAwMDEK"}}...]]>
            </certificateMetadata>
        </Certificate>
    </data>
</ServiceResponse>
```

Sample

Using a PEM file

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/certificate"

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

Request POST data

```
<ServiceRequest>
    <data>
        <Certificate>
            <name>Site01 Certificate</name>
            <description>A PEM certificate with ca-chain</description>
            <certificate><![CDATA[LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t
Ck1JSUdBRENDQStpZ0F3SUJBZ0lDRUFNdTTZxRXJpQWFFdS9SbGM4YmRjWnRUd0l0SwprM
0k4SUE9PQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==...]]></certificate>
            <privateKey><![CDATA[LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktL
S0tLQpQcm9jLVR5cGU6IDQsRU5DUllQVEVE9vdVVZanl0ZkxxUDRMdTdpbDVWMHlqNjIvc
XVqWmVBPT0KLS0tLS1FTkQgUlNBIFBSSVZBVEUgS0VZLS0tLS0K...]]></privateKey>
            <passphrase>furax</passphrase>
            <token>qualys</token>
            <chain><![CDATA[LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSU
YrRENDQStDZ0F3SUJBZ0lDRUFBd0RRWUpLb1pJaHZjTkFRRUxCUUF3Z1pFeEN6QUpCZ05W
QkFZVEFrWlMKTVE4d0RRWURWUVFJREVJUSUZJQ00tLS0tCg==...]]></chain>
        </Certificate>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/certificate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Certificate>
            <id>637</id>
            <uuid>9a3caf8c-498b-463b-84ce-4316af7454f7</uuid>
            <name>
                <![CDATA[Site01 Certificate]]>
            </name>
            <description>
```

```
                <![CDATA[A PEM certificate with ca-chain]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-04-06T08:31:34Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-04-06T08:58:01Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
                </updatedBy>
            <certificateMetadata>
                <![CDATA[{"fileName":null,"commonName":"site01.xfuente
s-docker","issuer":"EMAILADDRESS=xfuentes@qualys.com, CN=Intermediate
CA, OU=Engineering, O=Qualys, ST=France,
C=FR","dateStart":1486388496000,"dateEnd":1518788496000,"subject":"EMA
ILADDRESS=xfuentes@qualys.com, CN=site01.xfuentes-docker,
OU=Engineering, O=Qualys, L=Carcassonne, ST=France,
C=FR","sigAlgo":"SHA256WithRSAEncryption","sn":"4099","version":3,"isE
xpired":false,"isYetValid":true,"isSelfSigned":false,"publicKey":"UlNB
IFB1YmxpYyBLZXkKICAgICAgICAgICAgbW9kdWx1czogYTMyMTc1ZjI3YThiMjlkZGUyND
JhNjQ1ODIzNmFkMGQyMzFhZjc5MTQ5Mzk5MDIwY2FmYzM4YjE1ZGJhNzNmZWViYjY3Mz}.
..]]>
            </certificateMetadata>
            <chainMetadata>
                <![CDATA[[{"fileName":null,"commonName":"Intermediate
CA","issuer":"EMAILADDRESS=xfuentes@qualys.com, CN=Root CA,
OU=Engineering, O=Qualys, L=Carcassonne, ST=France,
C=FR","dateStart":1481796209000,"dateEnd":1797156209000,"subject":"EMA
ILADDRESS=xfuentes@qualys.com, CN=Intermediate CA, OU=Engineering,
O=Qualys, ST=France,
C=FR","sigAlgo":"SHA256WithRSAEncryption","sn":"4096","version":3,"isE
xpired":false,"isYetValid":true,"isSelfSigned":false,"publicKey":"UlNB
IFB1YmxpYyBLZXkKICAgICAgICAgICAgbW9kdWx1czogYmVkNmVhZWVlOTcxYzBkOWYxMz
```

```
g1M""xfuentes@qualys.com","issuerOrganization":"Qualys","publicKeySize
":1452,"privateKey":null}...]]]>
            </chainMetadata>
        </Certificate>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/certificate.xsd

Update certificate

/qps/rest/2.0/update/waf/certificate/<id>

[POST]

Update an SSL Certificate in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Update WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the SSL Certificate you want to update.

Optional input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	The name of the certificate profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the certificate profile.
pkcs12 (Text)	base64 encoded PFX file content (containing certificate and private key). The passphrase is required to decrypt the file.
certificate (Text)	base64 encoded PEM file content of the certificate (requires privateKey attribute).

privateKey (Text)	base64 encoded PEM file content of the private key. The passphrase is required if the key file is encrypted.
passphrase (Text)	Used to decrypt the provided PFX or PEM private key.
token (Text)	The privateKey of the certificate will be encrypted using this token. This token have to be specified on the WAF appliance where the certificate is to be installed to be able to decrypt it. Use the WAF appliance CLI to provided the token as the waf_ssl_passphrase.
chain (Text)	base64 encoded PEM file content of the certificate authority chain certificates.
tags	List of tags associated with the certificate profile. Valid action: Update

Sample

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/certificate/410"

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

Request POST data

```
<ServiceRequest>
    <data>
        <Certificate>
            <name>Site01 Certificate Updated</name>
            <description>A simple PEM certificate</description>
            <certificate><![CDATA[LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t
Ck1JSUdBRENDQStpZ0F3SUJBZ0lDRUFNd0RRWUpLb1pJaHZjTkFRRUxCUUF3Z1lNeEN6QU
pCZ05WQkFZVEFrWlM9OdDXeTArTlZJQ0FURS0tLS0tCg==...]]></certificate>
```

```
            <privateKey><![CDATA[LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktL
S0tLQpQc4QzRFRDE2MjM5RgooUKMT0KLS0IFBSSVZBVEUgS0tLS0K...]]></privateKe
y>
            <passphrase>furax</passphrase>
            <token>qualys</token>
        </Certificate>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/certificate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Certificate>
            <id>410</id>
            <uuid>8f6398d1-e333-4c5d-95d7-c81201df4d94</uuid>
            <name>
                <![CDATA[Site01 Certificate Updated]]>
            </name>
            <description>
                <![CDATA[A simple PEM certificate]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-04-05T08:53:20Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-04-05T15:39:36Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
```

```
                <lastname>Doe</lastname>
            </updatedBy>
            <certificateMetadata>
                <![CDATA[{"fileName":null,"commonName":"site01.xfuente
s-docker","issuer":"EMAILADDRESS=xfuentes@qualys.com, CN=Intermediate
CA, OU=Engineering, O=Qualys, ST=France,
C=FR","dateStart":1486388496000,"dateEnd":1518788496000,"subject":"EMA
ILADDRESS=xfuentes@qualys.com, CN=site01.xfuentes-docker,
OU=Engineering, O=Qualys, L=Carcassonne, ST=France,
C=FR","sigAlgo":"SHA256WithRSAEncryption","sn":"4099","version":3,"isE
xpired":false,"isYetValid":true,"isSelfSigned":false,NzNmZWViYjY3MYzBm
OGYxNjRkMDEzYTMwNDliNTVjNmFiNTg1YjhkOTU5Yjalys","subjectEmail":"xfuent
es@qualys.com","issuerOrganization":"Qualys","publicKeySize":768,"priv
ateKey":null...}]]>
            </certificateMetadata>
        </Certificate>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/certificate.xsd

Update certificates (bulk)

/qps/rest/2.0/update/waf/certificate

[POST]

Update multiple SSL Certificates in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Update WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search certificates.

Allowed input elements for bulk update are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Supported filter operators

Parameter	Description
name (Text)	The name of the certificate profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the certificate profile.
pkcs12 (Text)	base64 encoded PFX file content (containing certificate and private key). The passphrase is required to decrypt the file.
certificate (Text)	base64 encoded PEM file content of the certificate

	(requires privateKey attribute).
privateKey (Text)	base64 encoded PEM file content of the private key. The passphrase is required if the key file is encrypted.
passphrase (Text)	Used to decrypt the provided PFX or PEM private key.
token (Text)	The privateKey of the certificate will be encrypted using this token. This token have to be specified on the WAF appliance where the certificate is to be installed to be able to decrypt it. Use the WAF appliance CLI to provided the token as the waf_ssl_passphrase.
chain (Text)	base64 encoded PEM file content of the certificate authority chain certificates.
tags	List of tags associated with the certificate profile. Valid action: Update

Sample

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/certificate"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">qualys</Criteria>
    </filters>
    <data>
```

```
        <Certificate>
            <name>QCumber Certificate</name>
            <description>A simple test certificate
updated</description>
        </Certificate>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/certificate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Certificate>
            <id>637</id>
            <uuid>9a3caf8c-498b-463b-84ce-4316af7454f7</uuid>
            <name>
                <![CDATA[QCumber Certificate]]>
            </name>
            <description>
                <![CDATA[A simple test certificate updated]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-04-06T08:31:34Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-04-06T08:35:57Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
```

```
            </updatedBy>
            <certificateMetadata>
                <![CDATA[{"fileName":"uploaded-
cert.pfx","commonName":"site03.xfuentes-
docker","issuer":"EMAILADDRESS=xfuentes@qualys.com, CN=Intermediate
CA, OU=Engineering, O=Qualys, ST=France,
C=FR","dateStart":1490021391000,"dateEnd":1522421391000,"subject":"EMA
ILADDRESS=xfuentes@qualys.com, CN=site03.xfuentes-docker,
OU=Engineering, O=Qualys, L=Carcassonne, ST=France,
C=FR","sigAlgo":"SHA256WithRSAEncryption","sn":"4101","version":3,"isE
xpired":false,"isYetValid":true,"isSelfSigned":false,"publicKey":"UlNB
IFB1YmxpYyBLZXkKICAgICAgICAgICAgbW9kdWx1czogZTFkYWFiMmQ0NjNkOWQxZTIyNG
YwZmU1ZDUyYzY2NDQ1NWE4NzBmY2RhNWQyMzRjM2U3NWYxNGNiYjUwM2Q3ODVmZWY0Nj"s
ubjectEmail":"xfuentes@qualys.com","issuerOrganization":"Qualys","publ
icKeySize":768,"privateKey":null...}]]>
            </certificateMetadata>
        </Certificate>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/certificate.xsd

Delete certificate

/qps/rest/2.0/delete/waf/certificate/<id>

[POST]

Delete an SSL Certificate in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Delete WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the SSL Certificate you want to delete.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/delete/waf/certificate/637"
```

Response

```
<?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
/waf/certificate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Certificate>
            <id>637</id>
        </Certificate>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/certificate.xsd

Delete certificates (bulk)

/qps/rest/2.0/delete/waf/certificate

[POST]

Delete multiple SSL Certificates in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Delete WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search certificates.

Sample

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/delete/waf/certificate"

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

Request POST data

```
<?xml version="1.0" ?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">DEMO</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/certificate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Certificate>
            <id>638</id>
        </Certificate>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/certificate.xsd

Custom Response Pages API

Current custom response page count

/qps/rest/2.0/count/waf/custompage

[GET]

Returns the total number of custom response pages for WAF in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission and asset must be within the user's scope.

Input Parameters

No input elements are available.

Sample

API request

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

Response

```
<?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
/waf/custompage.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/custompage.xsd

Get details on a custom response page

/qps/rest/2.0/get/waf/custompage/<id>

[GET]

Returns details about a specific custom response page for WAF, within the user's scope. Want to find a custom response page ID to use as input? See Search custom response pages.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission and asset must be within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the custom response page ID of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/get/waf/custompage/1001"
```

Response

```
<?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
/waf/custompage.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <CustomPage>
            <id>1001</id>
            <uuid>11bf0ac8-3bde-4e10-aa75-ce4399378c58</uuid>
            <name>
                <![CDATA[my test page]]>
            </name>
```

```
            <description>
                <![CDATA[description]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-06-08T12:29:40Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-06-08T12:29:40Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <body>
                <![CDATA[
                <!DOCTYPE html>
                <html>
                    <body>
                        <h1>My Custom Page</h1>
                        <p>My custom content</p>
                    </body></html>]]>
                </body>
                <webApps>
                    <WebApp>
                        <id>63098273</id>
                        <uuid>01bd1b58-2802-48dd-b5b5-
ea1342aea21a</uuid>
                        <name>
                            <![CDATA[Site 01]]>
                        </name>
                    </WebApp>
                </webApps>
            </CustomPage>
        </data>
    </ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/custompage.xsd

Search custom response pages

/qps/rest/2.0/search/waf/custompage

[POST]

Finds custom response pages in the user's account matching the search criteria.

Permissions 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. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Supported filter operators

Parameter	Description
id (Long)	Custom response page identifier on Qualys Cloud Platform.
uuid (UUID)	Custom response page identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the custom response page as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the custom response page.
body (Text)	The body of the custom page (in HTML).

owner.id (Long)	The user ID of the custom response page owner.
owner.username (Text)	The user name of the custom response page owner.
owner.firstname (Text)	The first name of the custom response page owner.
owner.lastname (Text)	The last name of the custom response page owner.
created (Date)	The date/time when the custom response page was created.
createdBy.id (Long)	The user ID who created the custom response page.
createdBy.username (Text)	The user name who created the custom response page.
createdBy.firstname (Text)	The first name of the user who created the custom response page.
createdBy.lastname (Text)	The last name of the user who created the custom response page.
updated (Date)	The date/time when the custom response page was last updated.
updatedBy.id (Long)	The user ID who last updated the custom response page.
updatedBy.username (Text)	The user name who last updated the custom response page.
updatedBy.firstname (Text)	The first name of the user who updated the custom response page.
updatedBy.lastname (Text)	The last name of the user who updated the custom response page.

webApps.webApp.id (Long)	The ID of the Web Application this custom response page is associated with.
webApps.webApp.uuid (UUID)	The UUID of the Web Application this custom response page is associated with.
webApps.webApp.name (Text)	The name of the Web Application this custom response page is associated with.
tags.tag.id (Long)	The ID of a tag associated with the custom response page.
tags.tag.name (Text)	The name, defined by a user, of a tag associated with the custom response page.

Sample

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/custompage"

Note: "file.xml" contains the request POST data.
The request POST data is optional. If you leave it empty all custom
response pages in the user’s scope are returned.
```

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
   <filters>
      <Criteria field="id" operator="EQUALS">1601</Criteria>
   </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/custompage.xsd">
```

```
      <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <CustomPage>
            <id>1601</id>
            <uuid>8ebcf58c-9731-47b3-850e-8e20a2627f91</uuid>
            <name>
                <![CDATA[Custom unroutable]]>
            </name>
            <owner>
                <id>354401</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-07-28T00:01:19Z</created>
            <createdBy>
                <id>354401</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-07-28T18:31:31Z</updated>
            <updatedBy>
                <id>361390</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <body>
                <![CDATA[Wrong server has been hit.]]>
            </body>
            <webApps/>
        </CustomPage>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/custompage.xsd

Create custom response page

/qps/rest/2.0/create/waf/custompage

[POST]

Creates a custom response page which you can assign to a web application.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Create WAF Asset permissions.

Input Parameters

Allowed input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	(Required) The name of the custom response page as defined by a user. This is unique in subscription. Valid action: Update
body (Text)	(Required) The body of the custom page (in HTML).
description (Text)	The description of the custom response page.
tags	List of tags associated with the custom response page. Valid action: Update
tags.tag.id (Long)	The ID of a tag associated with the custom response page.
tags.tag.name	The name, defined by a user, of a tag associated with the custom response page.

(Text)

Sample

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/custompage"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <CustomPage>
            <name>my API created page 1</name>
            <description>example description</description>
            <body>
                <![CDATA[
                <!DOCTYPE html>
                <html>
                    <body>
                        <h1>My Custom Page</h1>
                        <p>My custom content</p>
                    </body></html>]]>
            </body>
        </CustomPage>
    </data>
</ServiceRequest>
```

Response

```
Response
<?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
/waf/custompage.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
```

```
        <CustomPage>
            <id>1401</id>
            <uuid>494d5aaa-2519-4e58-a6d4-699cfe2154a3</uuid>
            <name>
                <![CDATA[my API created page 1]]>
            </name>
            <description>
                <![CDATA[example description]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-06-12T10:00:28Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-06-12T10:00:28Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <body>
                <![CDATA[
                <!DOCTYPE html>
                <html>
                    <body>
                        <h1>My Custom Page</h1>
                        <p>My custom content</p>
                    </body></html>]]>
            </body>
            <webApps/>
        </CustomPage>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/custompage.xsd

Update custom response page

qps/rest/2.0/update/waf/custompage/<id>

[POST]

Update a custom response page in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Update WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the custom response page you want to update.

Optional input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	The name of the custom response page as defined by a user. This is unique in subscription. Valid action: Update
body (Text)	The body of the custom page (in HTML).
description (Text)	The description of the custom response page.
tags	List of tags associated with the custom response page. Valid action: Update

Sample

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/custompage/1401"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <CustomPage>
            <name>my API updated page 1</name>
            <description>updated description</description>
            <body>
                <![CDATA[
                <!DOCTYPE html>
                <html>
                    <body>
                        <h1>My Updated Custom Page</h1>
                        <p>My custom content</p>
                    </body></html>]]>
            </body>
        </CustomPage>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/custompage.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <CustomPage>
            <id>1401</id>
            <uuid>494d5aaa-2519-4e58-a6d4-699cfe2154a3</uuid>
            <name>
                <![CDATA[my API updated page 1]]>
```

```
            </name>
            <description>
                <![CDATA[updated description]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-06-12T10:00:28Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-06-12T10:12:59Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <body>
                <![CDATA[
                <!DOCTYPE html>
                <html>
                    <body>
                        <h1>My Updated Custom Page</h1>
                        <p>My custom content</p>
                    </body></html>]]>
            </body>
        </CustomPage>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/custompage.xsd

Update custom response pages (bulk)

qps/rest/2.0/update/waf/custompage

[POST]

Update a custom response page in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Update WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search custom response pages.

Allowed input elements for bulk update are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Supported filter operators

Parameter	Description
name (Text)	The name of the custom response page as defined by a user. This is unique in subscription. Valid action: Update
body (Text)	The body of the custom page (in HTML).
description (Text)	The description of the custom response page.
tags	List of tags associated with the custom response page.

Valid action: Update

Sample

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/custompage"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">API</Criteria>
  </filters>
  <data>
    <CustomPage>
        <description>bulk update</description>
    </CustomPage>
  </data>
</ServiceRequest>
```

Response

```
<?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
/waf/custompage.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <CustomPage>
            <id>1401</id>
            <uuid>494d5aaa-2519-4e58-a6d4-699cfe2154a3</uuid>
            <name>
                <![CDATA[my API updated page 1]]>
            </name>
            <description>
                <![CDATA[bulk update]]>
```

```
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </owner>
            <created>2017-06-12T10:00:28Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </createdBy>
            <updated>2017-06-12T10:29:25Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstname>John</firstname>
                <lastname>Doe</lastname>
            </updatedBy>
            <body>
                <![CDATA[
                <!DOCTYPE html>
                <html>
                    <body>
                        <h1>My Updated Custom Page</h1>
                        <p>My custom content</p>
                    </body></html>]]>
            </body>
            <webApps/>
        </CustomPage>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/custompage.xsd

Delete custom response page

/qps/rest/2.0/delete/waf/custompage/<id>

[POST]

Delete a custom response page in user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Delete WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the custom response page you want to delete.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/delete/waf/custompage/1202"
```

Response

```
<?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
/waf/custompage.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <CustomPage>
            <id>1202</id>
        </CustomPage>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/custompage.xsd

Delete custom response pages (bulk)

/qps/rest/2.0/delete/waf/custompage

[POST]

Delete multiple custom response pages in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS", Delete WAF Asset permissions, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search custom response pages.

Sample

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/delete/waf/custompage"

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

Request POST data

```
<?xml version="1.0" ?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">API</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/custompage.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <CustomPage>
            <id>1401</id>
        </CustomPage>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/custompage.xsd

Security Policies API

Current security policy count

/qps/rest/2.0/count/waf/securitypolicy

[GET]

Returns the total number of security policies for WAF in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and asset must be within the user's scope.

Input Parameters

No input elements are available.

Sample

API request

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

Response

```
<?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
/waf/securitypolicy.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>15</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/securitypolicy.xsd

Get details on a security policy

/qps/rest/2.0/get/waf/securitypolicy/<id>

[GET]

Returns details about a specific security policy for WAF, within the user's scope. Want to find a security policy ID to use as input? See Search security policies.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Access WAF module" permission, and asset must be within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the security policy of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/get/waf/securitypolicy/3348
1"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://lqualysapi.qualys.com/qps/xsd/2.
0/waf/securitypolicy.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <SecurityPolicy>
            <id>33481</id>
            <uuid>0fded90c-42da-4ae8-b5a5-998562a4990e</uuid>
            <name>
                <![CDATA[Server Security Policy]>
```

```
            </name>
            <description>
                <![CDATA[Security policies for servers]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </owner>
            <created>2017-05-14T12:33:20Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </createdBy>
            <updated>2017-05-14T12:33:22Z</updated>
            <updatedBy>
                <id>3988443</id>
               <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </updatedBy>
            <system>0</system>
            <applicationSecurity>
                <commandExecution>
                    <confidence>HIGH</confidence>
                </commandExecution>
                <crossSiteScripting>
                    <confidence>MEDIUM</confidence>
                </crossSiteScripting>
                <directoryTraversal>
                    <confidence>DISABLED</confidence>
                </directoryTraversal>
                <formatStringAttacks>
                    <confidence>LOW</confidence>
                </formatStringAttacks>
                <informationLeakage>
                    <value>46</value>
                </informationLeakage>
                <ldapInjection>
                    <value>47</value>
                </ldapInjection>
                <lfiAttacks>
```

```
                    <value>48</value>
                </lfiAttacks>
                <pathTraversal>
                    <value>49</value>
                </pathTraversal>
                <rfiAttacks>
                    <value>51</value>
                </rfiAttacks>
                <sourceCodeDisclosure>
                    <value>52</value>
                </sourceCodeDisclosure>
                <sqlInjection>
                    <value>53</value>
                </sqlInjection>
                <ssiInjection>
                    <value>54</value>
                </ssiInjection>
                <xpathInjection>
                    <value>55</value>
                </xpathInjection>
            </applicationSecurity>
            <threatLevel>
                <loggingThreshold>35</loggingThreshold>
                <blockingThreshold>65</blockingThreshold>
            </threatLevel>
        </SecurityPolicy>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/securitypolicy.xsd

Search security policies

/qps/rest/2.0/search/waf/securitypolicy

[POST]

Finds security policies in the user's account matching the search criteria.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and asset must be 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. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Supported filter operators

Parameter	Description
id (Long)	Security policy identifier on Qualys Cloud Platform.
uuid (UUID)	Security policy identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the security policy as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the security policy.
system (Integer)	Profile type: 0 if custom, 1 if system or 2 if template

owner.id (Long)	The user ID of the security policy owner.
owner.username (Text)	The user name of the security policy owner.
owner.firstname (Text)	The first name of the security policy owner.
owner.lastname (Text)	The last name of the security policy owner.
created (Date)	The date/time when the security policy was created.
createdBy.id (Long)	The user ID who created the security policy.
createdBy.username (Text)	The user name who created the security policy.
createdBy.firstname (Text)	The first name of the user who created the security policy.
createdBy.lastname (Text)	The last name of the user who created the security policy.
updated (Date)	The date/time when the security policy was last updated.
updatedBy.id (Long)	The user ID who last updated the security policy.
updatedBy.username (Text)	The user name who last updated the security policy.
updatedBy.firstname (Text)	The first name of the user who updated the security policy.
updatedBy.lastname (Text)	The last name of the user who updated the security policy.
webApps.webApp.id (Long)	The ID of the Web Application this security policy is associated with.

webApps.webApp.uuid (UUID)	The UUID of the Web Application this security policy is associated with.
webApps.webApp.name (Text)	The name of the Web Application this security policy is associated with.
tags.tag.id (Long)	The ID of a tag associated with the security policy.
tags.tag.name (Text)	The name, defined by a user, of a tag associated with the security policy.

Sample

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/securitypolicy"

Note: "file.xml" contains the request POST data.
The request POST data is optional. If you leave it empty all security
policies in the user’s scope are returned.
```

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <filters>
        <Criteria field="system" operator="EQUALS">1</Criteria>
    </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/securitypolicy.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <hasMoreRecords>false</hasMoreRecords>
```

```
    <data>
        <SecurityPolicy>
            <id>30681</id>
            <uuid>da1da5e5-7c2b-4a64-853b-651e41f2ed6d</uuid>
            <name>
                <![CDATA[Pass-through]]>
            </name>
            <owner>
                <id>3988442</id>
                 <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </owner>
            <created>2017-03-27T13:18:47Z</created>
            <createdBy>
                <id>3988442</id>
                 <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </createdBy>
            <updated>2017-03-27T13:18:47Z</updated>
            <system>1</system>
            <applicationSecurity>
                <commandExecution>
                    <confidence>DISABLED</confidence>
                </commandExecution>
                <crossSiteScripting>
                    <confidence>DISABLED</confidence>
                </crossSiteScripting>
                <directoryTraversal>
                    <confidence>DISABLED</confidence>
                </directoryTraversal>
                <formatStringAttacks>
                    <confidence>DISABLED</confidence>
                </formatStringAttacks>
                <informationLeakage>
                    <confidence>DISABLED</confidence>
                </informationLeakage>
                <ldapInjection>
                    <confidence>DISABLED</confidence>
                </ldapInjection>
                <lfiAttacks>
                    <confidence>DISABLED</confidence>
                </lfiAttacks>
                <pathTraversal>
```

```
                <confidence>DISABLED</confidence>
            </pathTraversal>
            <rfiAttacks>
                <confidence>DISABLED</confidence>
            </rfiAttacks>
            <sourceCodeDisclosure>
                <confidence>DISABLED</confidence>
            </sourceCodeDisclosure>
            <sqlInjection>
                <confidence>DISABLED</confidence>
            </sqlInjection>
            <ssiInjection>
                <confidence>DISABLED</confidence>
            </ssiInjection>
            <xpathInjection>
                <confidence>DISABLED</confidence>
            </xpathInjection>
        </applicationSecurity>
        <threatLevel>
            <loggingThreshold>50</loggingThreshold>
            <blockingThreshold>50</blockingThreshold>
        </threatLevel>
    </SecurityPolicy>
    <SecurityPolicy>
        <id>30682</id>
        <uuid>005e0d28-026c-49cc-9f40-87d5accac97f</uuid>
        <name>
            <![CDATA[Standard Policy]]>
        </name>
        <owner>
            <id>3988442</id>
            <username>john_doe</username>
            <firstName><John></firstName>
            <lastName><Doe></lastName>
        </owner>
        <created>2017-03-27T13:18:47Z</created>
        <createdBy>
            <id>3988442</id>
            < <username>john_doe</username>
            <firstName><John></firstName>
            <lastName><Doe></lastName>
        </createdBy>
        <updated>2017-03-27T13:18:47Z</updated>
        <system>1</system>
        <applicationSecurity>
```

```
                <commandExecution>
                    <confidence>MEDIUM</confidence>
                </commandExecution>
                <crossSiteScripting>
                    <confidence>MEDIUM</confidence>
                </crossSiteScripting>
                <directoryTraversal>
                    <confidence>MEDIUM</confidence>
                </directoryTraversal>
                 ...
                <rfiAttacks>
                    <confidence>MEDIUM</confidence>
                </rfiAttacks>
                <sourceCodeDisclosure>
                    <confidence>MEDIUM</confidence>
                </sourceCodeDisclosure>
                <sqlInjection>
                    <confidence>MEDIUM</confidence>
                </sqlInjection>
                <ssiInjection>
                    <confidence>MEDIUM</confidence>
                </ssiInjection>
                <xpathInjection>
                    <confidence>MEDIUM</confidence>
                </xpathInjection>
            </applicationSecurity>
            <threatLevel>
                <loggingThreshold>25</loggingThreshold>
                <blockingThreshold>75</blockingThreshold>
            </threatLevel>
        </SecurityPolicy>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/securitypolicy.xsd

Create security policy

/qps/rest/2.0/create/waf/securitypolicy

[POST]

Create a new security policy with given parameters.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and "Create Policy" permission.

Input Parameters

Allowed input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	(Required) The name of the security policy as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the security policy.
applicationSecurity	(Keyword: DISABLED, LOW, MEDIUM or HIGH) Used to specify the event confidence by name or by value (20-80) for all theses event types: commandExecution crossSiteScripting informationLeakage

	ldapInjection lfiAttacks rfiAttacks sourceCodeDisclosure sqlInjection ssiInjection xpathInjection rpo xmlInjection elInjection codeInjection
threatLevel.loggingThreshold (Integer)	Events with a confidence higher or equal than the specified value will be logged.
threatLevel.blockingThreshold (Integer)	Events with a confidence higher or equal than the specified value will be blocked.
tags	List of tags (identifier and name).
tags.tag.id (Long)	The ID of a tag associated with the security policy.
tags.tag.name (Text)	The name, defined by a user, of a tag associated with the security policy.

Sample

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/securitypolicy"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <SecurityPolicy>
            <name>Server Security Policy</name>
            <description>Security Policy for servers</description>
            <applicationSecurity>
                <commandExecution>
                    <confidence>HIGH</confidence>
                </commandExecution>
                <crossSiteScripting/>
                <directoryTraversal>
                    <confidence>DISABLED</confidence>
                </directoryTraversal>
                <formatStringAttacks>
                    <confidence>LOW</confidence>
                </formatStringAttacks>
                <informationLeakage>
                    <value>46</value>
                </informationLeakage>
                <ldapInjection>
                    <value>47</value>
                </ldapInjection>
                <lfiAttacks>
                    <value>48</value>
                </lfiAttacks>
                <pathTraversal>
                    <value>49</value>
                </pathTraversal>
                <rfiAttacks>
                    <value>51</value>
                </rfiAttacks>
                <sourceCodeDisclosure>
                    <value>52</value>
                </sourceCodeDisclosure>
                <sqlInjection>
                    <value>53</value>
                </sqlInjection>
                <ssiInjection>
```

```
                    <value>54</value>
                </ssiInjection>
                <xpathInjection>
                    <value>55</value>
                </xpathInjection>
            </applicationSecurity>
            <threatLevel>
                <loggingThreshold>35</loggingThreshold>
                <blockingThreshold>65</blockingThreshold>
            </threatLevel>
        </SecurityPolicy>
    </data>
</ServiceRequest>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/2.
0/waf/securitypolicy.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <SecurityPolicy>
            <id>33481</id>
            <uuid>0fded90c-42da-4ae8-b5a5-998562a4990e</uuid>
            <name>
                <![CDATA[Server Security Policy]]>
            </name>
            <description>
                <![CDATA[Security Policy for servers]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </owner>
            <created>2017-05-14T12:33:20Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </createdBy>
```

```
            <updated>2017-05-14T12:33:22Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </updatedBy>
            <system>0</system>
            <applicationSecurity>
                <commandExecution>
                    <confidence>HIGH</confidence>
                </commandExecution>
                <crossSiteScripting>
                    <confidence>MEDIUM</confidence>
                </crossSiteScripting>
                <directoryTraversal>
                    <confidence>DISABLED</confidence>
                </directoryTraversal>
                <formatStringAttacks>
                    <confidence>LOW</confidence>
                </formatStringAttacks>
                <informationLeakage>
                    <value>46</value>
                </informationLeakage>
                <ldapInjection>
                    <value>47</value>
                </ldapInjection>
                <lfiAttacks>
                    <value>48</value>
                </lfiAttacks>
                <pathTraversal>
                    <value>49</value>
                </pathTraversal>
                <rfiAttacks>
                    <value>51</value>
                </rfiAttacks>
                <sourceCodeDisclosure>
                    <value>52</value>
                </sourceCodeDisclosure>
                <sqlInjection>
                    <value>53</value>
                </sqlInjection>
                <ssiInjection>
                    <value>54</value>
                </ssiInjection>
```

```
                <xpathInjection>
                    <value>55</value>
                </xpathInjection>
            </applicationSecurity>
            <threatLevel>
                <loggingThreshold>35</loggingThreshold>
                <blockingThreshold>65</blockingThreshold>
            </threatLevel>
        </SecurityPolicy>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/securitypolicy.xsd

Update security policy

/qps/rest/2.0/update/waf/securitypolicy/<id>

[POST]

Update a security policy identified by its identifier with given parameters. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Update Policy" permission, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the security policy you want to update.

Optional input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	The name of the security policy as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the security policy.
applicationSecurity	(Keyword: DISABLED, LOW, MEDIUM or HIGH) Used to specify the event confidence by name or by value (20-80) for all theses event types:

	commandExecution crossSiteScripting informationLeakage ldapInjection lfiAttacks rfiAttacks sourceCodeDisclosure sqlInjection ssiInjection xpathInjection rpo xmlInjection elInjection codeInjection
threatLevel.loggingThreshold (Integer)	Events with a confidence higher or equal than the specified value will be logged.
threatLevel.blockingThreshold (Integer)	Events with a confidence higher or equal than the specified value will be blocked.
tags	List of tags (identifier and name).

Sample

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/securitypolicy"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <SecurityPolicy>
            <description>test policy updated</description>
            <applicationSecurity>
                <commandExecution>
                    <value>31</value>
                </commandExecution>
                <crossSiteScripting>
                    <value>32</value>
                </crossSiteScripting>
                <directoryTraversal>
                    <value>33</value>
                </directoryTraversal>
                <formatStringAttacks>
                    <value>34</value>
                </formatStringAttacks>
                <informationLeakage>
                    <value>35</value>
                </informationLeakage>
                <ldapInjection>
                    <value>36</value>
                </ldapInjection>
                <lfiAttacks>
                    <value>37</value>
                </lfiAttacks>
                <pathTraversal>
                    <value>38</value>
                </pathTraversal>
                <rfiAttacks>
                    <value>39</value>
                </rfiAttacks>
                <sourceCodeDisclosure>
                    <value>40</value>
                </sourceCodeDisclosure>
                <sqlInjection>
                    <value>41</value>
                </sqlInjection>
```

```
                <ssiInjection>
                    <value>42</value>
                </ssiInjection>
                <xpathInjection>
                    <value>43</value>
                </xpathInjection>
            </applicationSecurity>
            <threatLevel>
                <loggingThreshold>36</loggingThreshold>
                <blockingThreshold>64</blockingThreshold>
            </threatLevel>
        </SecurityPolicy>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/securitypolicy.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <SecurityPolicy>
            <id>33482</id>
            <uuid>68f06d15-9763-4df7-83e1-fbc9e3cdea4d</uuid>
            <name>
                <![CDATA[my security policy]]>
            </name>
            <description>
                <![CDATA[test policy updated]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </owner>
            <created>2017-05-14T13:07:27Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
```

```
            </createdBy>
            <updated>2017-05-14T13:10:44Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </updatedBy>
            <system>0</system>
            <applicationSecurity>
                <commandExecution>
                    <value>31</value>
                </commandExecution>
                <crossSiteScripting>
                    <value>32</value>
                </crossSiteScripting>
                <directoryTraversal>
                    <value>33</value>
                </directoryTraversal>
                <formatStringAttacks>
                    <value>34</value>
                </formatStringAttacks>
                <informationLeakage>
                    <value>35</value>
                </informationLeakage>
                <ldapInjection>
                    <value>36</value>
                </ldapInjection>
                <lfiAttacks>
                    <value>37</value>
                </lfiAttacks>
                <pathTraversal>
                    <value>38</value>
                </pathTraversal>
                <rfiAttacks>
                    <value>39</value>
                </rfiAttacks>
                <sourceCodeDisclosure>
                    <value>40</value>
                </sourceCodeDisclosure>
                <sqlInjection>
                    <value>41</value>
                </sqlInjection>
                <ssiInjection>
                    <value>42</value>
```

```
                </ssiInjection>
                <xpathInjection>
                    <value>43</value>
                </xpathInjection>
            </applicationSecurity>
            <threatLevel>
                <loggingThreshold>36</loggingThreshold>
                <blockingThreshold>64</blockingThreshold>
            </threatLevel>
        </SecurityPolicy>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/securitypolicy.xsd

Update security policies (bulk)

/qps/rest/2.0/update/waf/securitypolicy

[POST]

Update security policies identified by a search with given parameters. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Update Policy" permission, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search security policies.

Allowed input elements are listed below. The associated data type for each element appears in parentheses. All elements are optional.

Supported filter operators

Parameter	Description
name (Text)	The name of the security policy as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the security policy.
applicationSecurity	(Keyword: DISABLED, LOW, MEDIUM or HIGH) Used to specify the event confidence by name or by value (20-80) for all theses event types:

	commandExecution crossSiteScripting informationLeakage ldapInjection lfiAttacks rfiAttacks sourceCodeDisclosure sqlInjection ssiInjection xpathInjection rpo xmlInjection elInjection codeInjection
threatLevel.loggingThreshold (Integer)	Events with a confidence higher or equal than the specified value will be logged.
threatLevel.blockingThreshold (Integer)	Events with a confidence higher or equal than the specified value will be blocked.
tags	List of tags (identifier and name).

Sample

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/securitypolicy"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <filters>
        <Criteria field="system" operator="EQUALS">0</Criteria>
    </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/securitypolicy.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <SecurityPolicy>
            <id>31881</id>
            <uuid>d9c7aba3-5139-4553-a105-5e4a94eedd6d</uuid>
            <name>
                <![CDATA[Security Policy One]]>
            </name>
            <description>
                <![CDATA[Updating multiple security policies]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </owner>
            <created>2017-04-15T12:41:09Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </createdBy>
            <updated>2017-05-14T13:34:22Z</updated>
```

```
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </updatedBy>
            <system>0</system>
            <applicationSecurity>
                <commandExecution>
                    <confidence>MEDIUM</confidence>
                </commandExecution>
                <crossSiteScripting>
                    <value>71</value>
                </crossSiteScripting>
                <directoryTraversal>
                    <confidence>DISABLED</confidence>
                </directoryTraversal>
                <formatStringAttacks>
                    <confidence>DISABLED</confidence>
                </formatStringAttacks>
                <informationLeakage>
                    <value>36</value>
                </informationLeakage>
                <ldapInjection>
                    <confidence>DISABLED</confidence>
                </ldapInjection>
                <lfiAttacks>
                    <confidence>MEDIUM</confidence>
                </lfiAttacks>
                <pathTraversal>
                    <confidence>DISABLED</confidence>
                </pathTraversal>
                <rfiAttacks>
                    <confidence>MEDIUM</confidence>
                </rfiAttacks>
                <sourceCodeDisclosure>
                    <confidence>MEDIUM</confidence>
                </sourceCodeDisclosure>
                <sqlInjection>
                    <confidence>MEDIUM</confidence>
                </sqlInjection>
                <ssiInjection>
                    <confidence>MEDIUM</confidence>
                </ssiInjection>
                <xpathInjection>
```

```
                <confidence>MEDIUM</confidence>
            </xpathInjection>
        </applicationSecurity>
        <threatLevel>
            <loggingThreshold>23</loggingThreshold>
            <blockingThreshold>77</blockingThreshold>
        </threatLevel>
    </SecurityPolicy>
    <SecurityPolicy>
        <id>33482</id>
        <uuid>68f06d15-9763-4df7-83e1-fbc9e3cdea4d</uuid>
        <name>
            <![CDATA[Security Policy Two]]>
        </name>
        <description>
            <![CDATA[Updating multiple security policies]]>
        </description>
        <owner>
            <id>3988443</id>
            <username>john_doe</username>
            <firstName><John></firstName>
            <lastName><Doe></lastName>
        </owner>
        <created>2017-05-14T13:07:27Z</created>
        <createdBy>
            <id>3988443</id>
            <username>john_doe</username>
            <firstName><John></firstName>
            <lastName><Doe></lastName>
        </createdBy>
        <updated>2017-05-14T13:34:34Z</updated>
        <updatedBy>
            <id>3988443</id>
            <username>john_doe</username>
            <firstName><John></firstName>
            <lastName><Doe></lastName>
        </updatedBy>
        <system>0</system>
        <applicationSecurity>
            <commandExecution>
                <value>31</value>
            </commandExecution>
            <crossSiteScripting>
                <value>32</value>
            </crossSiteScripting>
```

```
                <directoryTraversal>
                    <value>33</value>
                </directoryTraversal>
                <formatStringAttacks>
                    <value>34</value>
                </formatStringAttacks>
                <informationLeakage>
                    <value>35</value>
                </informationLeakage>
                <ldapInjection>
                    <value>36</value>
                </ldapInjection>
                <lfiAttacks>
                    <value>37</value>
                </lfiAttacks>
                <pathTraversal>
                    <value>38</value>
                </pathTraversal>
                <rfiAttacks>
                    <value>39</value>
                </rfiAttacks>
                <sourceCodeDisclosure>
                    <value>40</value>
                </sourceCodeDisclosure>
                <sqlInjection>
                    <value>41</value>
                </sqlInjection>
                <ssiInjection>
                    <value>42</value>
                </ssiInjection>
                <xpathInjection>
                    <value>43</value>
                </xpathInjection>
            </applicationSecurity>
            <threatLevel>
                <loggingThreshold>23</loggingThreshold>
                <blockingThreshold>77</blockingThreshold>
            </threatLevel>
        </SecurityPolicy>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/securitypolicy.xsd

Delete security policy

/qps/rest/2.0/delete/waf/securitypolicy/<id>

[POST]

Delete an existing security policy identified by its identifier.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Delete Policy" permission, and asset must be within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the security policy of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/delete/waf/securitypolicy/3
2882"
```

Response

```
<?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
/waf/securitypolicy.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <SecurityPolicy>
            <id>32882</id>
        </SecurityPolicy>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/securitypolicy.xsd

Delete Custom Rules (bulk)

/qps/rest/2.0/delete/waf/securitypolicy/

[POST]

Delete security policies identified by a search with given parameters.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Delete Policy" permission, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search security policies.

Sample

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/delete/waf/securitypolicy"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">security
policy</Criteria>
    </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/securitypolicy.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>4</count>
    <data>
        <SecurityPolicy>
            <id>32681</id>
        </SecurityPolicy>
        <SecurityPolicy>
            <id>32881</id>
        </SecurityPolicy>
        <SecurityPolicy>
            <id>33083</id>
        </SecurityPolicy>
        <SecurityPolicy>
            <id>33481</id>
        </SecurityPolicy>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/securitypolicy.xsd

HTTP Profiles API

Current HTTP Profile count

/qps/rest/2.0/count/waf/httpprofile

[GET]

Returns the total number of HTTP profile count for WAF in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and asset must be within the user's scope.

Input Parameters

No input elements are available.

Sample

API request

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

Response

```
<?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
/waf/httpprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>4</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/httpprofile.xsd

Get details on an HTTP Profile

/qps/rest/2.0/get/waf/httpprofile/<id>

[GET]

Returns details about a specific HTTP profile for WAF, within the user's scope. Want to find an HTTP profile ID to use as input? See Search HTTP Profiles.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and asset must be within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the HTTP profile of interest. The associated data type for each element appears in parentheses. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Parameter	Description
id (Long)	HTTP profile identifier on Qualys Cloud Platform.
uuid (UUID)	HTTP profile identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the HTTP profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the HTTP profile.
system (Boolean)	True if this is a system profile.

urls.string (Text)	At least one backend web server address (HTTP URL).
requestMethod	Request Method protection settings.
keepAcceptEncoding (Boolean)	If true, the Accept Encoding header field will be kept in the request.
requestHeader	Request Header protection settings.
requestContentType	Request Content Type protection settings (allowAll or denyAll sub elements).
detectProtocolAnomalies (Boolean)	Enable/disable protocol anomalies detection.
webServiceProtection.xmlParsing.enabled (Boolean)	XML parser enabled or disabled. Default is disabled.
webServiceProtection.xmlParsing.size (Integer)	Maximum size of data parsed (in bytes). Note that the size of data can get inflated due to pattern reuse (e.g. Reuse of XML entities). Size also includes extra payloads added for preventing against attacks. For example to prevent a Billion laughs attack (a DoS attack aimed at XML parsers). Default is 100000 characters.
webServiceProtection.xmlParsing.items (Integer)	Maximum number of items parsed. An "item" can be an attribute, element tag, etc. (Depending on format: whether XML or JSON). Default is 10000.

webServiceProtection.xmlParsing.level (Integer)	Maximum depth reachable when parsing structured content. This enables you to avoid parsing data with huge depth, but protects servers against DDOS attacks. Default is 32.
webServiceProtection.jsonParsing.enabled (Boolean)	JSON parser enabled or disabled. Default is disabled.
webServiceProtection.jsonParsing.size (Integer)	Maximum size of data parsed (in bytes). Note that the size of data can get inflated due to pattern reuse (e.g. Reuse of XML entities). Size also includes extra payloads added for preventing against attacks. For example to prevent a Billion laughs attack (a DoS attack aimed at XML parsers). Default is 100000 characters.
webServiceProtection.jsonParsing.items (Integer)	Maximum number of items parsed. An "item" can be an attribute, element tag, etc. (Depending on format: whether XML or JSON). Default is 10000.
webServiceProtection.jsonParsing.level (Integer)	Maximum depth reachable when parsing structured content. This enables you to avoid parsing data with huge depth, but protects servers against DDOS attacks. Default is 32.
serverCloaking	Server Cloaking settings.
suppressSensitiveHeaders (Boolean)	Suppress sensitive headers if true.
onErrorMessages (Keyword)	Action when error messages are

	returned (ALLOW, LOG, BLOCK).
onSensitiveFileTypes (Keyword)	Action when sensitive file types are requested (ALLOW, LOG, BLOCK).
onSensitiveFileExtensions (Keyword)	Action when sensitive file extensions are requested (ALLOW, LOG, BLOCK).
cookieProtection	Cookie protection settings.
discourageContentTypeSniffing (Boolean)	If true discourage content type sniffing.
forceDefaultContentType (Text)	Force default content type when unknown (enabled attribute and cdata value).
forceDefaultCharacterEncoding (Text)	Force default character encoding (type attribute and cdata value).
contentSecurityPolicyHeader (Text)	Content security policy header (enabled attribute and cdata value).
discourageClickjacking (Keyword)	Discourage click jacking (NONE, NO_FRAMING, SAME_ORIGIN_FRAMING).
browserXSSProtection (Keyword)	Protect browser from XSS attacks (NONE, DISABLE, ENABLE_WITHOUT_BLOCKING, ENABLE_WITH_BLOCKING).
owner (Text)	The user for Qualys Cloud Platform who owns this HTTP profile.

owner.id (Long)	The user ID of the HTTP profile owner.
owner.username (Text)	The user name of the HTTP profile owner.
owner.firstname (Text)	The first name of the HTTP profile owner.
owner.lastname (Text)	The last name of the HTTP profile owner.
created (Date)	The date/time when the HTTP profile was created.
createdBy.id (Long)	The user ID who created the HTTP profile.
createdBy.username (Text)	The user name who created the HTTP profile.
createdBy.firstname (Text)	The first name of the user who created the HTTP profile.
createdBy.lastname (Text)	The last name of the user who created the HTTP profile.
updated (Date)	The date/time when the HTTP profile was last updated.
updatedBy.id (Long)	The user ID who last updated the HTTP profile.
updatedBy.username (Text)	The user name who last updated the HTTP profile.
updatedBy.firstname (Text)	The first name of the user who updated the HTTP profile.
updatedBy.lastname (Text)	The last name of the user who

	updated the HTTP profile.
tags	List of tags (identifier and name).
tags.tag.id (Long)	A tag identifier in tag list of that HTTP profile.
tags.tag.name (Text)	A tag name in tag list of that HTTP profile.

Sample

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"
```

Response

```
<?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
/waf/httpprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <HTTPProfile>
            <id>4401</id>
            <uuid>4ffd8f5e-529d-4a38-9cac-7fb805962a18</uuid>
            <name>
                <![CDATA[My HTTP Profile]]>
            </name>
            <description>
                <![CDATA[My first HTTP 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>
                <detectInvalid>false</detectInvalid>
                <detectRepeated>false</detectRepeated>
                <detectChunked>false</detectChunked>
            </requestHeader>
            <requestContentType/>
<detectProtocolAnomalies>false</detectProtocolAnomalies>
         <webServiceProtection>
                <xmlParsing>
                    <enabled>true</enabled>
                    <size>500000</size>
                    <items>50000</items>
                    <level>64</level>
                </xmlParsing>
                <jsonParsing>
                    <enabled>true</enabled>
                    <size>600000</size>
                    <items>10000</items>
                    <level>32</level>
                </jsonParsing>
            </webServiceProtection>
            <serverCloaking>
                <enabled>false</enabled>
            </serverCloaking>
<suppressSensitiveHeaders>true</suppressSensitiveHeaders>
            <onErrorMessages>BLOCK</onErrorMessages>
            <onSensitiveFileTypes>BLOCK</onSensitiveFileTypes>
<onSensitiveFileExtensions>BLOCK</onSensitiveFileExtensions>
```

```
            <cookieProtection>
                <type>NONE</type>
            </cookieProtection>
<discourageContentTypeSniffing>false</discourageContentTypeSniffing>
            <forceDefaultContentType>
                <enabled>false</enabled>
            </forceDefaultContentType>
            <forceDefaultCharacterEncoding>
                <type>NONE</type>
            </forceDefaultCharacterEncoding>
            <contentSecurityPolicyHeader>
                <enabled>false</enabled>
            </contentSecurityPolicyHeader>
<discourageClickjacking>SAME_ORIGIN_FRAMING</discourageClickjacking>
<browserXSSProtection>ENABLE_WITH_BLOCKING</browserXSSProtection>
        </HTTPProfile>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/httpprofile.xsd

Search HTTP Profiles

/qps/rest/2.0/search/waf/httpprofile

[POST]

Finds HTTP profiles in the user's account matching the search criteria.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and asset must be 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. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Supported filter operators

Parameter	Description
id (Long)	HTTP profile identifier on Qualys Cloud Platform.
uuid (UUID)	HTTP profile identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the HTTP profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the HTTP profile.
system (Boolean)	True if this is a system profile.

owner.id (Long)	The user ID of the HTTP profile owner.
owner.username (Text)	The user name of the HTTP profile owner.
owner.firstname (Text)	The first name of the HTTP profile owner.
owner.lastname (Text)	The last name of the HTTP profile owner.
created (Date)	The date/time when the HTTP profile was created.
createdBy.id (Long)	The user ID who created the HTTP profile.
createdBy.username (Text)	The user name who created the HTTP profile.
createdBy.firstname (Text)	The first name of the user who created the HTTP profile.
createdBy.lastname (Text)	The last name of the user who created the HTTP profile.
updated (Date)	The date/time when the HTTP profile was last updated.
updatedBy.id (Long)	The user ID who last updated the HTTP profile.
updatedBy.username (Text)	The user name who last updated the HTTP profile.
updatedBy.firstname (Text)	The first name of the user who updated the HTTP profile.
updatedBy.lastname (Text)	The last name of the user who updated the HTTP profile.
webApps.webApp.id (Long)	The ID of the Web Application this HTTP profile is associated with.

webApps.webApp.uuid (UUID)	The UUID of the Web Application this HTTP profile is associated with.
webApps.webApp.name (Text)	The name of the Web Application this HTTP profile is associated with.
tags.tag.id (Long)	A tag identifier in tag list of that HTTP profile.
tags.tag.name (Text)	A tag name in tag list of that HTTP profile.

Sample

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/httpprofile"

Note: "file.xml" contains the request POST data.
The request POST data is optional. If you leave it empty all HTTP
profiles in the user’s scope are returned.
```

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
   <filters>
      <Criteria field="owner.id" operator="EQUALS">2501190</Criteria>
   </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/httpprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <HTTPProfile>
```

```
            <id>11801</id>
            <uuid>7e3c59a8-0dd1-4483-a136-22f2e8498a84</uuid>
            <name>
                <![CDATA[Standard Protocol]]>
            </name>
            <owner>
                <id>2501190</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </owner>
            <created>2017-03-01T22:22:28Z</created>
            <createdBy>
                <id>2501190</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </createdBy>
            <updated>2017-03-01T22:22:28Z</updated>
            <system>true</system>
            <requestMethod>
                <allowAll>
                    <detectInvalid>false</detectInvalid>
                    <detectTraceTrack>false</detectTraceTrack>
                </allowAll>
            </requestMethod>
            <keepAcceptEncoding>false</keepAcceptEncoding>
            <requestHeader>
                <detectInvalid>false</detectInvalid>
                <detectRepeated>false</detectRepeated>
                <detectChunked>false</detectChunked>
            </requestHeader>
            <requestContentType>
                <allowAll>
                    <detectFileUploads>false</detectFileUploads>
                </allowAll>
            </requestContentType>
            <detectProtocolAnomalies>false</detectProtocolAnomalies>
            <webServiceProtection>
                <xmlParsing>
                    <enabled>true</enabled>
                    <size>500000</size>
                    <items>50000</items>
                    <level>64</level>
                </xmlParsing>
```

```
                <jsonParsing>
                    <enabled>true</enabled>
                    <size>600000</size>
                    <items>10000</items>
                    <level>32</level>
                </jsonParsing>
            </webServiceProtection>
            <serverCloaking>
                <enabled>false</enabled>
            </serverCloaking>
            <suppressSensitiveHeaders>false</suppressSensitiveHeaders>
            <onErrorMessages>LOG</onErrorMessages>
            <onSensitiveFileTypes>LOG</onSensitiveFileTypes>
            <onSensitiveFileExtensions>LOG</onSensitiveFileExtensions>
            <cookieProtection>
                <type>NONE</type>
            </cookieProtection>
            <discourageContentTypeSniffing>false</discourageContentTyp
eSniffing>
            <forceDefaultContentType>
                <enabled>false</enabled>
            </forceDefaultContentType>
            <forceDefaultCharacterEncoding>
                <type>NONE</type>
            </forceDefaultCharacterEncoding>
            <contentSecurityPolicyHeader>
                <enabled>false</enabled>
            </contentSecurityPolicyHeader>
            <discourageClickjacking>NONE</discourageClickjacking>
            <browserXSSProtection>DISABLE</browserXSSProtection>
        </HTTPProfile>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/httpprofile.xsd

Create HTTP Profile

/qps/rest/2.0/create/waf/httpprofile

[POST]

Create a new HTTP profile with given parameters.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and "Create WAF Asset" permission.

Input Parameters

Allowed input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	(Required) The name of the HTTP profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the HTTP profile.
requestMethod.allowAll (Boolean)	(Required) Activate AllowAll Policy for HTTP Methods. (one of allowAll or denyAll is required)
requestMethod.denyAll (Boolean)	(Required) Activate DenyAll Policy for HTTP Methods except the one listed in the Cdata

	content. (one of allowAll or denyAll is required)
keepAcceptEncoding (Boolean)	If true, the Accept Encoding header field will be kept in the request. This parameter is set to false by default.
requestMethod.allowAll.detectTraceTrack (Boolean)	if true, enables trace track detection.
requestHeader	(Required) Request Header protection settings.
requestHeader.detectInvalid (Boolean)	If true enables invalid headers detection.
requestHeader.detectRepeated (Boolean)	If true enables repeated headers detection.
requestHeader.detectChunked (Boolean)	If true enables chunked headers detection.
requestContentType.allowAll (Boolean)	(Required) Activate AllowAll Policy for request Content Types. (one of allowAll or denyAll is required)
requestContentType.denyAll (Boolean)	(Required) Activate DenyAll Policy for request Content Types except for the one listed in the Cdata content. (one of allowAll or denyAll is required)
requestContentType.allowAll.detectFileUploads (Boolean)	If true enables file uploads detection.
detectProtocolAnomalies (Boolean)	(Required) Enable/disable protocol anomalies detection.
webServiceProtection.xmlParsing.enabled	(Required) XML parser enabled

(Boolean)	or disabled. Default is disabled.
webServiceProtection.xmlParsing.size (Integer)	Maximum size of data parsed (in bytes). Note that the size of data can get inflated due to pattern reuse (e.g. Reuse of XML entities). Size also includes extra payloads added for preventing against attacks. For example to prevent a Billion laughs attack (a DoS attack aimed at XML parsers). Default is 100000 characters.
webServiceProtection.xmlParsing.items (Integer)	Maximum number of items parsed. An "item" can be an attribute, element tag, etc. (Depending on format: whether XML or JSON). Default is 10000.
webServiceProtection.xmlParsing.level (Integer)	Maximum depth reachable when parsing structured content. This enables you to avoid parsing data with huge depth, but protects servers against DDOS attacks. Default is 32.
webServiceProtection.jsonParsing.enabled (Boolean)	(Required) JSON parser enabled or disabled. Default is disabled.
webServiceProtection.jsonParsing.size (Integer)	Maximum size of data parsed (in bytes). Note that the size of data can get inflated due to pattern reuse (e.g. Reuse of XML entities). Size also includes extra payloads added for preventing against attacks. For example to prevent a Billion laughs attack (a DoS attack aimed at XML parsers). Default is 100000 characters.

webServiceProtection.jsonParsing.items (Integer)	Maximum number of items parsed. An "item" can be an attribute, element tag, etc. (Depending on format: whether XML or JSON). Default is 10000.
webServiceProtection.jsonParsing.level (Integer)	Maximum depth reachable when parsing structured content. This enables you to avoid parsing data with huge depth, but protects servers against DDOS attacks. Default is 32.
serverCloaking	(Required) Server Cloaking settings.
serverCloaking.enabled (Boolean)	If true enable server Cloaking.
serverCloaking.value (Text)	(Required) Use to specify the server header value that will be enforced.
suppressSensitiveHeaders (Boolean)	(Required) Suppress sensitive headers if true.
onErrorMessages (Keyword)	(Required) Action when error messages are returned (ALLOW, LOG, BLOCK).
onSensitiveFileTypes (Keyword)	(Required) Action when sensitive file types are requested (ALLOW, LOG, BLOCK).
onSensitiveFileExtensions (Keyword)	(Required) Action when sensitive file extensions are requested (ALLOW, LOG, BLOCK).
cookieProtection	(Required) Cookie protection settings.

cookieProtection.type (Keyword)	Cookie protection type (NONE, ALL, SELECTED).
cookieProtection.value (Text)	Use to specify the list of selected cookies if cookie protection type is "SELECTED".
discourageContentTypeSniffing (Boolean)	(Required) If true discourage content type sniffing.
forceDefaultContentType (Text)	Force default content type when unknown (enabled attribute and cdata value).
forceDefaultContentType.enabled (Boolean)	If true forces default content type when not specified.
forceDefaultContentType.value (Text)	(Required) Use to specify the default content type to enforce.
forceDefaultCharacterEncoding (Text)	(Required) Force default character encoding (type attribute and cdata value).
forceDefaultCharacterEncoding.type (Keyword)	Type of enforcement (NONE, ALWAYS_APPLY, APPLY_WHEN_NOT_SET).
forceDefaultCharacterEncoding.value (Text)	Use to specify the default character encoding to enforce.
contentSecurityPolicyHeader (Text)	(Required) Content security policy header (enabled attribute and cdata value).
contentSecurityPolicyHeader.enabled (Boolean)	If true enables content security header.
contentSecurityPolicyHeader.value (Text)	(Required) Use to specify the value of the content security

	header.
discourageClickjacking (Keyword)	(Required) Discourage click jacking (NONE, NO_FRAMING, SAME_ORIGIN_FRAMING).
browserXSSProtection (Keyword)	(Required) Protect browser from XSS attacks (NONE, DISABLE, ENABLE_WITHOUT_BLOCKING, ENABLE_WITH_BLOCKING).
tags	List of tags (identifier and name).
tags.tag.id (Long)	A tag identifier in tag list of that HTTP profile.
tags.tag.name (Text)	A tag name in tag list of that HTTP profile.

Sample

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>HTTP profile for server</name>
            <description>HTTP profile created for
servers</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>
            <requestContentType>
                <allowAll>
                    <detectFileUploads>false</detectFileUploads>
                </allowAll>
            </requestContentType>
            <onErrorMessages>BLOCK</onErrorMessages>
            <onSensitiveFileTypes>BLOCK</onSensitiveFileTypes>
    <onSensitiveFileExtensions>BLOCK</onSensitiveFileExtensions>
            <cookieProtection>
                <type>NONE</type>
            </cookieProtection>
         <discourageContentTypeSniffing>true</discourageContentTypeSni
ffing>
            <forceDefaultContentType>
                 <enabled>false</enabled>
            </forceDefaultContentType>
            <forceDefaultCharacterEncoding>
                <type>NONE</type>
            </forceDefaultCharacterEncoding>
            <contentSecurityPolicyHeader>
                <enabled>false</enabled>
            </contentSecurityPolicyHeader>
            <discourageClickjacking>SAME_ORIGIN_FRAMING</discourageCli
ckjacking>
            <browserXSSProtection>ENABLE_WITH_BLOCKING</browserXSSProt
ection>
            <serverCloaking>
                <enabled>false</enabled>
            </serverCloaking>
            <suppressSensitiveHeaders>true</suppressSensitiveHeaders>
        </HTTPProfile>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/httpprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <HTTPProfile>
            <id>4801</id>
            <uuid>d6f60c16-5146-477d-a005-a2d182cb0632</uuid>
            <name>
                <![CDATA[HTTP profile for server]]>
            </name>
            <description>
                <![CDATA[HTTP profile created for servers]]>
            </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>
            <requestContentType>
                <allowAll>
                    <detectFileUploads>false</detectFileUploads>
                </allowAll>
            </requestContentType>
            <detectProtocolAnomalies>false</detectProtocolAnomalies>
            <webServiceProtection>
                <xmlParsing>
                    <enabled>false</enabled>
                </xmlParsing>
                <jsonParsing>
                    <enabled>false</enabled>
                </jsonParsing>
            </webServiceProtection>
            <serverCloaking>
                <enabled>false</enabled>
            </serverCloaking>
            <suppressSensitiveHeaders>true</suppressSensitiveHeaders>
            <onErrorMessages>BLOCK</onErrorMessages>
            <onSensitiveFileTypes>BLOCK</onSensitiveFileTypes>
            <onSensitiveFileExtensions>BLOCK</onSensitiveFileExtension
s>
            <cookieProtection>
                <type>NONE</type>
            </cookieProtection>
            <discourageContentTypeSniffing>true</discourageContentType
Sniffing>
            <forceDefaultContentType>
                <enabled>false</enabled>
            </forceDefaultContentType>
            <forceDefaultCharacterEncoding>
                <type>NONE</type>
            </forceDefaultCharacterEncoding>
            <contentSecurityPolicyHeader>
                <enabled>false</enabled>
            </contentSecurityPolicyHeader>
            <discourageClickjacking>SAME_ORIGIN_FRAMING</discourageCli
ckjacking>
            <browserXSSProtection>ENABLE_WITH_BLOCKING</browserXSSProt
ection>
```

```
        </HTTPProfile>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/httpprofile.xsd

Update HTTP Profile

/qps/rest/2.0/update/waf/httpprofile/<id>

[POST]

Update a HTTP profile identified by its identifier with given parameters. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Update WAF Asset" permission, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the HTTP profile you want to update.

Optional input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	The name of the HTTP profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the HTTP profile.
requestMethod.allowAll (Boolean)	Activate AllowAll Policy for HTTP Methods. (one of allowAll or denyAll is required)

requestMethod.denyAll (Boolean)	Activate DenyAll Policy for HTTP Methods except the one listed in the Cdata content. (one of allowAll or denyAll is required)
keepAcceptEncoding (Boolean)	If true, the Accept Encoding header field will be kept in the request. This parameter is set to false by default.
requestMethod.allowAll.detectTraceTrack (Boolean)	if true enables trace track detection.
requestHeader	Request Header protection settings.
requestHeader.detectInvalid (Boolean)	If true enables invalid headers detection.
requestHeader.detectRepeated (Boolean)	If true enables repeated headers detection.
requestHeader.detectChunked (Boolean)	If true enables chunked headers detection.
requestContentType.allowAll (Boolean)	Activate AllowAll Policy for request Content Types. (one of allowAll or denyAll is required)
requestContentType.denyAll (Boolean)	Activate DenyAll Policy for request Content Types except for the one listed in the Cdata content. (one of allowAll or denyAll is required)
requestContentType.allowAll.detectFileUploads (Boolean)	If true enables file uploads detection.

detectProtocolAnomalies (Boolean)	Enable/disable protocol anomalies detection.
webServiceProtection.xmlParsing.enabled (Boolean)	XML parser enabled or disabled. Default is disabled.
webServiceProtection.xmlParsing.size (Integer)	Maximum size of data parsed (in bytes). Note that the size of data can get inflated due to pattern reuse (e.g. Reuse of XML entities). Size also includes extra payloads added for preventing against attacks. For example to prevent a Billion laughs attack (a DoS attack aimed at XML parsers). Default is 100000 characters.
webServiceProtection.xmlParsing.items (Integer)	Maximum number of items parsed. An "item" can be an attribute, element tag, etc. (Depending on format: whether XML or JSON). Default is 10000.
webServiceProtection.xmlParsing.level (Integer)	Maximum depth reachable when parsing structured content. This enables you to avoid parsing data with huge depth, but protects servers against DDOS attacks. Default is 32.
webServiceProtection.jsonParsing.enabled (Boolean)	JSON parser enabled or disabled. Default is disabled.
webServiceProtection.jsonParsing.size (Integer)	Maximum size of data parsed (in bytes). Note that the size of data can get inflated due to pattern reuse (e.g. Reuse of XML entities). Size also includes extra payloads added for preventing against attacks. For

	example to prevent a Billion laughs attack (a DoS attack aimed at XML parsers). Default is 100000 characters.
webServiceProtection.jsonParsing.items (Integer)	Maximum number of items parsed. An "item" can be an attribute, element tag, etc. (Depending on format: whether XML or JSON). Default is 10000.
webServiceProtection.jsonParsing.level (Integer)	Maximum depth reachable when parsing structured content. This enables you to avoid parsing data with huge depth, but protects servers against DDOS attacks. Default is 32.
serverCloaking	Server Cloaking settings.
serverCloaking.enabled (Boolean)	If true enable server Cloaking.
serverCloaking.value (Text)	Use to specify the server header value that will be enforced.
suppressSensitiveHeaders (Boolean)	Suppress sensitive headers if true.
onErrorMessages (Keyword)	Action when error messages are returned (ALLOW, LOG, BLOCK).
onSensitiveFileTypes (Keyword)	Action when sensitive file types are requested (ALLOW, LOG, BLOCK).

onSensitiveFileExtensions (Keyword)	Action when sensitive file extensions are requested (ALLOW, LOG, BLOCK).
cookieProtection	Cookie protection settings.
cookieProtection.type (Keyword)	Cookie protection type (NONE, ALL, SELECTED).
cookieProtection.value (Text)	Use to specify the list of selected cookies if cookie protection type is "SELECTED".
discourageContentTypeSniffing (Boolean)	If true discourage content type sniffing.
forceDefaultContentType (Text)	Force default content type when unknown (enabled attribute and cdata value).
forceDefaultContentType.enabled (Boolean)	If true forces default content type when not specified.
forceDefaultContentType.value (Text)	Use to specify the default content type to enforce.
forceDefaultCharacterEncoding (Text)	Force default character encoding (type attribute and cdata value).
forceDefaultCharacterEncoding.type (Keyword)	Type of enforcement (NONE, ALWAYS_APPLY, APPLY_WHEN_NOT_SET).
forceDefaultCharacterEncoding.value (Text)	Use to specify the default character encoding to enforce.
contentSecurityPolicyHeader (Text)	Content security policy header (enabled attribute and cdata

	value).
contentSecurityPolicyHeader.enabled (Boolean)	If true enables content security header.
contentSecurityPolicyHeader.value (Text)	Use to specify the value of the content security header.
discourageClickjacking (Keyword)	Discourage click jacking (NONE, NO_FRAMING, SAME_ORIGIN_FRAMING).
browserXSSProtection (Keyword)	Protect browser from XSS attacks (NONE, DISABLE, ENABLE_WITHOUT_BLOCKING, ENABLE_WITH_BLOCKING).
tags	List of tags (identifier and name).

Sample

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/4801
"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <HTTPProfile>
            <description>Update HTTP profile description</description>
            <forceDefaultContentType>
                <enabled>true</enabled>
```

```
                <value><![CDATA[application/xml]]></value>
            </forceDefaultContentType>
<discourageClickjacking>NO_FRAMING</discourageClickjacking>
        </HTTPProfile>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/httpprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <HTTPProfile>
            <id>4801</id>
            <uuid>d6f60c16-5146-477d-a005-a2d182cb0632</uuid>
            <name>
                <![CDATA[HTTP Profile]]>
            </name>
            <description>
                <![CDATA[Update HTTP profile description]]>
            </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>2017-05-04T12:40:22Z</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>
            </requestHeader>
            <requestContentType>
                <allowAll>
                    <detectFileUploads>false</detectFileUploads>
                </allowAll>
            </requestContentType>
<detectProtocolAnomalies>false</detectProtocolAnomalies>
             <webServiceProtection>
                <xmlParsing>
                    <enabled>true</enabled>
                    <size>500000</size>
                    <items>50000</items>
                    <level>64</level>
                </xmlParsing>
                <jsonParsing>
                    <enabled>true</enabled>
                    <size>600000</size>
                    <items>10000</items>
                    <level>32</level>
                </jsonParsing>
            </webServiceProtection>
            <serverCloaking>
                <enabled>false</enabled>
            </serverCloaking>
<suppressSensitiveHeaders>true</suppressSensitiveHeaders>
            <onErrorMessages>BLOCK</onErrorMessages>
            <onSensitiveFileTypes>BLOCK</onSensitiveFileTypes>
<onSensitiveFileExtensions>BLOCK</onSensitiveFileExtensions>
            <cookieProtection>
                <type>NONE</type>
            </cookieProtection>
<discourageContentTypeSniffing>true</discourageContentTypeSniffing>
            <forceDefaultContentType>
```

```
                <enabled>true</enabled>
                <value>
                    <![CDATA[application/xml]]>
                </value>
            </forceDefaultContentType>
            <forceDefaultCharacterEncoding>
                <type>NONE</type>
            </forceDefaultCharacterEncoding>
            <contentSecurityPolicyHeader>
                <enabled>false</enabled>
            </contentSecurityPolicyHeader>
<discourageClickjacking>NO_FRAMING</discourageClickjacking>
<browserXSSProtection>ENABLE_WITH_BLOCKING</browserXSSProtection>
        </HTTPProfile>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/httpprofile.xsd

Update HTTP Profiles (bulk)

/qps/rest/2.0/update/waf/httpprofile

[POST]

Update HTTP profiles identified by a search with given parameters. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Update WAF Asset" permission, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search HTTP Profiles.

Allowed input elements are listed below. The associated data type for each element appears in parentheses. All elements are optional.

Supported filter operators

Parameter	Description
name (Text)	The name of the HTTP profile as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the HTTP profile.
requestMethod.allowAll (Boolean)	Activate AllowAll Policy for HTTP Methods. (one of allowAll

	or denyAll is required)
requestMethod.denyAll (Boolean)	Activate DenyAll Policy for HTTP Methods except the one listed in the Cdata content. (one of allowAll or denyAll is required)
keepAcceptEncoding (Boolean)	If true, the Accept Encoding header field will be kept in the request. This parameter is set to false by default.
requestMethod.allowAll.detectTraceTrack (Boolean)	if true enables trace track detection.
requestHeader	Request Header protection settings.
requestHeader.detectInvalid (Boolean)	If true enables invalid headers detection.
requestHeader.detectRepeated (Boolean)	If true enables repeated headers detection.
requestHeader.detectChunked (Boolean)	If true enables chunked headers detection.
requestContentType.allowAll (Boolean)	Activate AllowAll Policy for request Content Types. (one of allowAll or denyAll is required)
requestContentType.denyAll (Boolean)	Activate DenyAll Policy for request Content Types except for the one listed in the Cdata content. (one of allowAll or denyAll is required)

requestContentType.allowAll.detectFileUploads (Boolean)	If true enables file uploads detection.
detectProtocolAnomalies (Boolean)	Enable/disable protocol anomalies detection.
webServiceProtection.xmlParsing.enabled (Boolean)	XML parser enabled or disabled. Default is disabled.
webServiceProtection.xmlParsing.size (Integer)	Maximum size of data parsed (in bytes). Note that the size of data can get inflated due to pattern reuse (e.g. Reuse of XML entities). Size also includes extra payloads added for preventing against attacks. For example to prevent a Billion laughs attack (a DoS attack aimed at XML parsers). Default is 100000 characters.
webServiceProtection.xmlParsing.items (Integer)	Maximum number of items parsed. An "item" can be an attribute, element tag, etc. (Depending on format: whether XML or JSON). Default is 10000.
webServiceProtection.xmlParsing.level (Integer)	Maximum depth reachable when parsing structured content. This enables you to avoid parsing data with huge depth, but protects servers against DDOS attacks. Default is 32.
webServiceProtection.jsonParsing.enabled (Boolean)	JSON parser enabled or disabled. Default is disabled.

webServiceProtection.jsonParsing.size (Integer)	Maximum size of data parsed (in bytes). Note that the size of data can get inflated due to pattern reuse (e.g. Reuse of XML entities). Size also includes extra payloads added for preventing against attacks. For example to prevent a Billion laughs attack (a DoS attack aimed at XML parsers). Default is 100000 characters.
webServiceProtection.jsonParsing.items (Integer)	Maximum number of items parsed. An "item" can be an attribute, element tag, etc. (Depending on format: whether XML or JSON). Default is 10000.
webServiceProtection.jsonParsing.level (Integer)	Maximum depth reachable when parsing structured content. This enables you to avoid parsing data with huge depth, but protects servers against DDOS attacks. Default is 32.
serverCloaking	Server Cloaking settings.
serverCloaking.enabled (Boolean)	If true enable server Cloaking.
serverCloaking.value (Text)	Use to specify the server header value that will be enforced.
suppressSensitiveHeaders (Boolean)	Suppress sensitive headers if true.
onErrorMessages (Keyword)	Action when error messages are returned (ALLOW, LOG, BLOCK).

onSensitiveFileTypes (Keyword)	Action when sensitive file types are requested (ALLOW, LOG, BLOCK).
onSensitiveFileExtensions (Keyword)	Action when sensitive file extensions are requested (ALLOW, LOG, BLOCK).
cookieProtection	Cookie protection settings.
cookieProtection.type (Keyword)	Cookie protection type (NONE, ALL, SELECTED).
cookieProtection.value (Text)	Use to specify the list of selected cookies if cookie protection type is "SELECTED".
discourageContentTypeSniffing (Boolean)	If true discourage content type sniffing.
forceDefaultContentType (Text)	Force default content type when unknown (enabled attribute and cdata value).
forceDefaultContentType.enabled (Boolean)	If true forces default content type when not specified.
forceDefaultContentType.value (Text)	Use to specify the default content type to enforce.
forceDefaultCharacterEncoding (Text)	Force default character encoding (type attribute and cdata value).
forceDefaultCharacterEncoding.type (Keyword)	Type of enforcement (NONE, ALWAYS_APPLY, APPLY_WHEN_NOT_SET).

forceDefaultCharacterEncoding.value (Text)	Use to specify the default character encoding to enforce.
contentSecurityPolicyHeader (Text)	Content security policy header (enabled attribute and cdata value).
contentSecurityPolicyHeader.enabled (Boolean)	If true enables content security header.
contentSecurityPolicyHeader.value (Text)	Use to specify the value of the content security header.
discourageClickjacking (Keyword)	Discourage click jacking (NONE, NO_FRAMING, SAME_ORIGIN_FRAMING).
browserXSSProtection (Keyword)	Protect browser from XSS attacks (NONE, DISABLE, ENABLE_WITHOUT_BLOCKING, ENABLE_WITH_BLOCKING).
tags	List of tags (identifier and name).

Sample

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>
  <filters>
```

```
    <Criteria field="description"
operator="CONTAINS">updated</Criteria>
  </filters>
  <data>
    <HTTPProfile>
        <description>Update description</description>
        <requestHeader>
            <detectInvalid>true</detectInvalid>
            <detectRepeated>true</detectRepeated>
            <detectChunked>true</detectChunked>
        </requestHeader>
    </HTTPProfile>
  </data>
</ServiceRequest>
```

Response

```
<?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
/waf/httpprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <HTTPProfile>
            <id>4801</id>
            <uuid>d6f60c16-5146-477d-a005-a2d182cb0632</uuid>
            <name>
                <![CDATA[HTTP Profile]]>
            </name>
            <description>
                <![CDATA[Update description]]>
            </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>2017-05-04T12:58:03Z</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>true</detectInvalid>
                <detectRepeated>true</detectRepeated>
                <detectChunked>true</detectChunked>
            </requestHeader>
            <requestContentType>
                <allowAll>
                    <detectFileUploads>false</detectFileUploads>
                </allowAll>
            </requestContentType>
<detectProtocolAnomalies>false</detectProtocolAnomalies>
            <webServiceProtection>
                <xmlParsing>
                    <enabled>true</enabled>
                    <size>500000</size>
                    <items>50000</items>
                    <level>64</level>
                </xmlParsing>
                <jsonParsing>
                    <enabled>true</enabled>
                    <size>600000</size>
                    <items>10000</items>
                    <level>32</level>
                </jsonParsing>
            </webServiceProtection>
            <serverCloaking>
                <enabled>false</enabled>
            </serverCloaking>
<suppressSensitiveHeaders>true</suppressSensitiveHeaders>
```

```
            <onErrorMessages>BLOCK</onErrorMessages>
            <onSensitiveFileTypes>BLOCK</onSensitiveFileTypes>
<onSensitiveFileExtensions>BLOCK</onSensitiveFileExtensions>
            <cookieProtection>
                <type>NONE</type>
            </cookieProtection>
<discourageContentTypeSniffing>true</discourageContentTypeSniffing>
            <forceDefaultContentType>
                <enabled>true</enabled>
                <value>
                    <![CDATA[application/xml]]>
                </value>
            </forceDefaultContentType>
            <forceDefaultCharacterEncoding>
                <type>NONE</type>
            </forceDefaultCharacterEncoding>
            <contentSecurityPolicyHeader>
                <enabled>false</enabled>
            </contentSecurityPolicyHeader>
<discourageClickjacking>NO_FRAMING</discourageClickjacking>
<browserXSSProtection>ENABLE_WITH_BLOCKING</browserXSSProtection>
        </HTTPProfile>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/httpprofile.xsd

Delete HTTP Profile

/qps/rest/2.0/delete/waf/httpprofile/<id>

[POST]

Delete an existing HTTP profile identified by its identifier.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Delete WAF Asset" permission, and asset must be within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the HTTP profile of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/delete/waf/httpprofile/1401
"
```

Response

```
<?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
/waf/httpprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <HTTPProfile>
            <id>1401</id>
        </HTTPProfile>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/httpprofile.xsd

Delete HTTP Profiles (bulk)

/qps/rest/2.0/delete/waf/httpprofile/

[POST]

Delete multiple HTTP profiles identified by search operation.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Delete WAF Asset" permission, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search HTTP Profiles.

Sample

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/delete/waf/httpprofile"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
 <ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">API</Criteria>
    </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/httpprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>3</count>
    <data>
        <HTTPProfile>
            <id>4401</id>
        </HTTPProfile>
        <HTTPProfile>
            <id>4601</id>
        </HTTPProfile>
        <HTTPProfile>
            <id>4801</id>
        </HTTPProfile>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/httpprofile.xsd

Custom Rules API

Current cluster count

/qps/rest/2.0/count/waf/customrule

[GET]

Returns the total number of custom rules for WAF in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and asset must be within the user's scope.

Input Parameters

No input elements are available.

Sample

API request

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

Response

```
<?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
/waf/customrule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/customrule.xsd

Get details on a Custom Rule

/qps/rest/2.0/get/waf/customrule/<id>

[GET]

Returns details about a specific custom rule for WAF, within the user's scope. Want to find a custom rule ID to use as input? See Search Custom Rules.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and asset must be within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the custom rule of interest. The associated data type for each element appears in parentheses. All dates must be entered in UTC date/time format.

Parameter	Description
id (Long)	Custom rule identifier on Qualys Cloud Platform.
uuid (UUID)	Custom rule identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the custom rule as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the custom rule.
conditions (Text)	Used to specify the conditions to meet for this rule to be activated. Each RuleCondition is composed of a subject an operator and a value. Optionally a key can be provided for subjects that allow specifying a

	custom key like a special header or parmeter name.
action (Text)	The action to execute if the rule conditions were met. Can be block, allow or redirect, or block with custom page. Additionally a log element can be added with a value of true to enable logging of the action. In case of redirect a code and url element needs to be provided with a valid HTTP Redirect code (302, 301) and a valid HTTP URL. In case of block with custom page, provide the ID, UUID or name element to identify the custom page to be associated.
owner.id (Long)	The user ID of the custom rule owner.
owner.username (Text)	The user name of the custom rule owner.
owner.firstname (Text)	The first name of the custom rule owner.
owner.lastname (Text)	The last name of the custom rule owner.
created (Date)	The date/time when the custom rule was created.
createdBy.id (Long)	The user ID who created the custom rule.
createdBy.username (Text)	The user name who created the custom rule.
createdBy.firstname (Text)	The first name of the user who created the custom rule.
createdBy.lastname (Text)	The last name of the user who created the custom rule.

updated (Date)	The date/time when the custom rule was last updated.
updatedBy.id (Long)	The user ID who last updated the custom rule.
updatedBy.username (Text)	The user name who last updated the custom rule.
updatedBy.firstname (Text)	The first name of the user who updated the custom rule.
updatedBy.lastname (Text)	The last name of the user who updated the custom rule.
tags	Tags associated with this custom rule.
tags.tag.id (Long)	A tag identifier in tag list of that custom rule.
tags.tag.name (Text)	A tag name in tag list of that custom rule.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/get/waf/customrule/1001"
```

Response

```
<?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
/waf/customrule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <CustomRule>
            <id>1001</id>
            <uuid>183186b8-4c91-490c-90e0-1bdf4752f3fe</uuid>
            <name>
                <![CDATA[Custom Rule Server]]>
```

```
            </name>
            <description>
                <![CDATA[Description for Custom Rule]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </owner>
            <created>2017-04-14T13:57:32Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </createdBy>
            <updated>2017-05-16T07:32:28Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </updatedBy>
            <conditions>
                <RuleCondition>
                    <subject>
                        <![CDATA[request.header]]>
                    </subject>
                    <key>
                        <![CDATA[Secret]]>
                    </key>
                    <operator>EQUAL</operator>
                    <value>
                        <![CDATA[Qualys]]>
                    </value>
                </RuleCondition>
                <RuleCondition>
                    <subject>
                        <![CDATA[client.ip.address]]>
                    </subject>
                    <operator>EQUAL</operator>
                    <value>
                        <![CDATA[truc]]>
                    </value>
```

```
                </RuleCondition>
            </conditions>
            <action>
                <log>true</log>
                <block/>
            </action>
        </CustomRule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/customrule.xsd

Search Custom Rules

/qps/rest/2.0/search/waf/customrule

[POST]

Finds custom rules in the user's account matching the search criteria.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and asset must be 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. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Supported filter operators

Parameter	Description
id (Long)	Custom rule identifier on Qualys Cloud Platform.
uuid (UUID)	Custom rule identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the custom rule as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the custom rule.
owner.id (Long)	The user ID of the custom rule owner.
owner.username	The user name of the custom rule owner.

(Text)

owner.firstname (Text)	The first name of the custom rule owner.
owner.lastname (Text)	The last name of the custom rule owner.
created (Date)	The date/time when the custom rule was created.
createdBy.id (Long)	The user ID who created the custom rule.
createdBy.username (Text)	The user name who created the custom rule.
createdBy.firstname (Text)	The first name of the user who created the custom rule.
createdBy.lastname (Text)	The last name of the user who created the custom rule.
updated (Date)	The date/time when the custom rule was last updated.
updatedBy.id (Long)	The user ID who last updated the custom rule.
updatedBy.username (Text)	The user name who last updated the custom rule.
updatedBy.firstname (Text)	The first name of the user who updated the custom rule.
updatedBy.lastname (Text)	The last name of the user who updated the custom rule.
tags.tag.id (Long)	A tag identifier in tag list of that custom rule.
tags.tag.name (Text)	A tag name in tag list of that custom rule.

Sample

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/customrule"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">cust</Criteria>
    </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/customrule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <CustomRule>
            <id>1001</id>
            <uuid>183186b8-4c91-490c-90e0-1bdf4752f3fe</uuid>
            <name>
                <![CDATA[Custom Rule Servers]]>
            </name>
            <description>
                <![CDATA[My custom rule]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </owner>
            <created>2017-04-14T13:57:32Z</created>
```

```
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </createdBy>
            <updated>2017-05-16T07:32:28Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </updatedBy>
            <conditions>
                <RuleCondition>
                    <subject>
                        <![CDATA[request.header]]>
                    </subject>
                    <key><![CDATA[Secret]]></key>
                    <operator>EQUAL</operator>
                    <value>
                        <![CDATA[Qualys]]>
                    </value>
                </RuleCondition>
                <RuleCondition>
                    <subject>
                        <![CDATA[client.ip.address]]>
                    </subject>
                    <operator>EQUAL</operator>
                    <value><![CDATA[truc]]></value>
                </RuleCondition>
            </conditions>
            <action>
                <log>true</log>
                <block/>
            </action>
        </CustomRule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/customrule.xsd

Create Custom Rule

/qps/rest/2.0/create/waf/customrule

[POST]

Create a new custom rule with given parameters.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and "Create Patch/Exception Rule" permission.

Input Parameters

Allowed input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	(Required) The name of the custom rule as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the custom rule.
conditions (Text)	(Required) Used to specify the conditions to meet for this rule to be activated. Each RuleCondition is composed of a subject an operator and a value. Optionally a key can be provided for subjects that allow specifying a custom key like a special header or parmeter name. See Rule Conditions

action (Text)	(Required) The action to execute if the conditions were met. Can be block,allow, redirect, block with custom page or header related rules, such as insertHeader, rewriteHeader or stripHeader. In case of redirect a code and url element needs to be provided with a valid HTTP Redirect code (302, 301) and a valid HTTP URL. In case of block with custom page, provide the ID, UUID orname element to identify the custom page to be associated. In case of insertHeader, provide the name and value of the HTTP header to be added in the response. You can add a security header which instructs the browser exactly how to behave when it handles your website's content and data. An example of a security header could be an XFO header to mitigate clickjacking attacks: x-frame-options:SAMEORIGIN. In case of rewriteHeader/stripHeader, provide the name and value of the HTTP header to be modified and/or stripped in the response.
tags	Provide a list of existing tags to be assigned to this custom rule. Multiple tags can be provided in the form of a comma separated list.
tags.tag.id (Long)	A tag identifier in tag list of that custom rule.
tags.tag.name (Text)	A tag name in tag list of that custom rule.

Sample

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/customrule"
```

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <CustomRule>
            <name>My First Custom Rule</name>
            <description>checking servers</description>
            <conditions>
                <RuleCondition>
                    <subject>request.body.parameter</subject>
                    <key>blague</key>
                    <operator>EQUAL</operator>
                    <value>toto</value>
                </RuleCondition>
            </conditions>
            <action>
                <log>true</log>
                <block/>
            </action>
        </CustomRule>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/customrule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <CustomRule>
            <id>3001</id>
            <uuid>ade93c5d-12f6-4929-8e94-3132c1507f57</uuid>
            <name>
                <![CDATA[My First Custom Rule]]>
            </name>
            <description>
                <![CDATA[checking servers]]>
            </description>
```

```
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </owner>
            <created>2017-05-17T11:55:30Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </createdBy>
            <updated>2017-05-17T11:55:30Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
            </updatedBy>
            <conditions>
                <RuleCondition>
                    <subject>
                        <![CDATA[request.body.parameter]]>
                    </subject>
                    <key><![CDATA[blague]]></key>
                    <operator>EQUAL</operator>
                    <value>
                        <![CDATA[toto]]>
                    </value>
                </RuleCondition>
            </conditions>
            <action>
                <log>true</log>
                <block/>
            </action>
        </CustomRule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/customrule.xsd

Update Custom Rule

/qps/rest/2.0/update/waf/customrule/<id>

[POST]

Update a custom rule identified by its identifier with given parameters. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Update Patch/Exception Rule" permission, and asset must be within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the custom rule you want to update.

Optional input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	The name of the custom rule as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the custom rule.
conditions (Text)	Used to specify the conditions to meet for this rule to be activated. Each RuleCondition is composed of a subject an operator and a value. Optionally a key can be provided for subjects that allow specifying a custom key like a

	special header or parmeter name. See Rule Conditions.
action (Text)	(Required) The action to execute if the conditions were met. Can be block,allow, redirect, block with custom page or header related rules, such as insertHeader, rewriteHeader or stripHeader. In case of redirect a code and url element needs to be provided with a valid HTTP Redirect code (302, 301) and a valid HTTP URL. In case of block with custom page, provide the ID, UUID orname element to identify the custom page to be associated. In case of insertHeader, provide the name and value of the HTTP header to be added in the response. You can add a security header which instructs the browser exactly how to behave when it handles your website's content and data. An example of a security header could be an XFO header to mitigate clickjacking attacks: x-frame-options:SAMEORIGIN. In case of rewriteHeader/stripHeader, provide the name and value of the HTTP header to be modified and/or stripped in the response.
tags	Provide a list of existing tags to be assigned to this custom rule. Multiple tags can be provided in the form of a comma separated list.

Sample

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/customrule"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">rule</Criteria>
    </filters>
    <data>
        <CustomRule>
            <description>for qualys</description>
        </CustomRule>
    </data>
</ServiceRequest>
```

Response

```
<?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
/waf/customrule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <CustomRule>
            <id>2001</id>
            <uuid>84561e39-bc85-45cd-b8f1-c483c913750a</uuid>
            <name>
                <![CDATA[rule1]]>
            </name>
            <description>
                <![CDATA[for qualys]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
             </owner>
            <created>2017-04-19T07:45:17Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
             </createdBy>
            <updated>2017-05-17T12:16:34Z</updated>
```

```
            <updatedBy>
                <id>3988443</id>
               <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
             </updatedBy>
            <conditions>
                <RuleCondition>
                    <subject>
                        <![CDATA[request.body.charset]]>
                    </subject>
                    <operator>EQUAL</operator>
                    <value>
                        <![CDATA[UTF-16]]>
                    </value>
                </RuleCondition>
                <RuleCondition>
                    <subject>
                        <![CDATA[server.ip.address]]>
                    </subject>
                    <operator>EQUAL</operator>
                    <value>
                        <![CDATA[1.1.1.1]]>
                    </value>
                </RuleCondition>
                <RuleCondition>
                    <subject>
                        <![CDATA[request.header]]>
                    </subject>
                    <key>
                        <![CDATA[Qualys]]>
                    </key>
                    <operator>EQUAL</operator>
                    <value>
                        <![CDATA[Rox!]]>
                    </value>
                </RuleCondition>
            </conditions>
            <action>
                <redirect/>
                <code>302</code>
                <url>http://example.com</url>
            </action>
        </CustomRule>
        <CustomRule>
```

```
            <id>3001</id>
            <uuid>ade93c5d-12f6-4929-8e94-3132c1507f57</uuid>
            <name>
                <![CDATA[my test rule updated]]>
            </name>
            <description>
                <![CDATA[for qualys]]>
            </description>
            <owner>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
             </owner>
            <created>2017-05-17T11:55:30Z</created>
            <createdBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
             </createdBy>
            <updated>2017-05-17T12:16:38Z</updated>
            <updatedBy>
                <id>3988443</id>
                <username>john_doe</username>
                <firstName><John></firstName>
                <lastName><Doe></lastName>
             </updatedBy>
            <conditions>
                <RuleCondition>
                    <subject>
                        <![CDATA[request.header]]>
                    </subject>
                    <key>
                        <![CDATA[blague]]>
                    </key>
                    <operator>EQUAL</operator>
                    <value>
                        <![CDATA[toto]]>
                    </value>
                </RuleCondition>
            </conditions>
            <action>
                <log>true</log>
                <allow/>
```

```
            </action>
        </CustomRule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/customrule.xsd

Update Custom Rules (bulk)

/qps/rest/2.0/update/waf/customrule

[POST]

Update custom rules identified by a search with given parameters. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Update Patch/Exception Rule" permission, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search Custom Rules.

Allowed input elements are listed below. The associated data type for each element appears in parentheses. All elements are optional.

Supported filter operators

Parameter	Description
name (Text)	The name of the custom rule as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	A description of the custom rule.
conditions (Text)	Used to specify the conditions to meet for this rule to be activated. Each RuleCondition is composed of a subject an operator and a value. Optionally a key can be provided for subjects that allow specifying a custom key like a

	special header or parmeter name. See Rule Conditions.
action (Text)	(Required) The action to execute if the conditions were met. Can be block,allow, redirect, block with custom page or header related rules, such as insertHeader, rewriteHeader or stripHeader. In case of redirect a code and url element needs to be provided with a valid HTTP Redirect code (302, 301) and a valid HTTP URL. In case of block with custom page, provide the ID, UUID orname element to identify the custom page to be associated. In case of insertHeader, provide the name and value of the HTTP header to be added in the response. You can add a security header which instructs the browser exactly how to behave when it handles your website's content and data. An example of a security header could be an XFO header to mitigate clickjacking attacks: x-frame-options:SAMEORIGIN. In case of rewriteHeader/stripHeader, provide the name and value of the HTTP header to be modified and/or stripped in the response.
tags	Provide a list of existing tags to be assigned to this custom rule. Multiple tags can be provided in the form of a comma separated list.

Sample

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/customrule"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="createdBy.firstname"
operator="EQUALS">John</Criteria>
  </filters>
  <data>
    <Cluster>
      <tags><Tag><name>XYZ</name></Tag></tags>
    </Cluster>
  </data>
</ServiceRequest>
```

Response

```
<?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/waf/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Cluster>
      <id>25401</id>
      <uuid>7de38215-a9e3-4855-8566-d9b369a92756</uuid>
      <name><![CDATA[Montlegun Updated]]></name>
      <description><![CDATA[Manage all Appliances from
Montlegun]]></description>
      <errorResponse>
        <redirect>
          <url>https://www.domain.com/resturl.html</url>
          <status>302</status>
        </redirect>
      </errorResponse>
      <trustedIPs>
        <string>1.2.3.5</string>
        <string>1.2.3.6/23</string>
      </trustedIPs>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,THU</weekDays>
        <startTime>23</startTime>
        <timezone>
            <code>Asia/Aden</code>
```

```
            <offset>+03:00</offset>
          </timezone>
          <freezeEndDate>2018-02-28</freezeEndDate>
          <nextUpgradeDate>2018-03-01T23:00:00Z</nextUpgradeDate>
        </updateSchedule>
        <owner>
          <id>3989443</id>
          <username>john_doe</username>
          <firstname>John</firstname>
          <lastname>Doe</lastname>
        </owner>
        <created>2017-07-25T09:35:01Z</created>
        <createdBy>
          <id>3989443</id>
          <username>john_doe</username>
          <firstname>John</firstname>
          <lastname>Doe</lastname>
        </createdBy>
        <updated>2017-07-25T12:30:24Z</updated>
        <updatedBy>
          <id>3989444</id>
          <username>john_doe</username>
          <firstname>John</firstname>
          <lastname>Doe</lastname>
        </updatedBy>
        <token><![CDATA[C190F3A8-08FA-46DD-BB16-EB14ECC47606]]></token>
        <syncDate>2017-07-25T12:27:31Z</syncDate>
        <status>DEGRADED</status>
        <appliances>
          <Appliance>
            <id>15802</id>
            <uuid>580c5d0f-b80b-452d-a345-0ee1526fe88f</uuid>
            <name><![CDATA[580C5D0F-B80B-452D-A345-
0EE1526FE88F]]></name>
          </Appliance>
          <Appliance>
            <id>15803</id>
            <uuid>6c5844e9-86f5-418e-90d5-b0201123bf13</uuid>
            <name><![CDATA[6C5844E9-86F5-418E-90D5-
B0201123BF13]]></name>
          </Appliance>
        </appliances>
        <tags>
          <Tag>
            <id>10256059</id>
```

```
          <name><![CDATA[XYZ]]></name>
        </Tag>
      </tags>
    </Cluster>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/customrule.xsd

Delete Custom Rules

/qps/rest/2.0/delete/waf/customrule/<id>

[POST]

Delete an existing custom rule identified by its identifier.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Delete Patch/Exception Rule" permission, and asset must be within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the custom rule of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/delete/waf/customrule/1001"
```

Response

```
<?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
/waf/customrule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <CustomRule>
            <id>1001</id>
        </CustomRule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/customrule.xsd

Delete Custom Rules (bulk)

/qps/rest/2.0/delete/waf/customrule/

[POST]

Delete custom rules identified by a search with given parameters.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Delete Patch/Exception Rule" permission, and asset must be within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search Custom Rules.

Sample

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/delete/waf/customrule"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">rule</Criteria>
    </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/customrule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <CustomRule>
            <id>2001</id>
        </CustomRule>
        <CustomRule>
            <id>3001</id>
        </CustomRule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/customrule.xsd

Rule conditions

Custom Rule API supports the following conditions and operators for custom rule.

client.ip.address

(CIDR) IP address of the client. MATCH and NOT.MATCH accept regexp values.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH

Example

Match client ip address 172.26.10.123

```
<RuleCondition>
    <subject>
    <![CDATA[client.ip.address]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[172.26.10.123]>
    </value>
</RuleCondition>
```

client.ip.geolocation

(List) Country code of the client IP address

EQUAL, NOT.EQUAL

Example

Match any origin country except US

```
<RuleCondition>
    <subject>
    <![CDATA[client.ip.geolocation]]>
    </subject>
    <operator>NOT.EQUAL</operator>
```

```
    <value>
    <![CDATA[US]]>
    </value>
</RuleCondition>
```

client.ip.protocol

(List) Version of the client IP protocol, ipv4/ipv6

EQUAL, NOT.EQUAL

Example

Match any IPV4 source-ip address

```
<RuleCondition>
    <subject>
    <![CDATA[client.ip.protocol]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[IPV4]]>
    </value>
</RuleCondition>
```

client.ssl.cipher

(String) SSL cipher used by the client SSL connection. MATCH and NOT.MATCH accept regexp values.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH

Example

Match client-side ssl/tls cipher ECDHE-RSA-AES128-SHA256

```
<RuleCondition>
    <subject>
    <![CDATA[client.ssl.cipher]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[ECDHE-RSA-AES128-SHA256]]>
    </value>
</RuleCondition>
```

client.ssl.protocol

(List) Client SSL protocol used.

EQUAL, NOT.EQUAL

Example

Match sslv2 protocol on client-side

```
<RuleCondition>
    <subject>
    <![CDATA[client.ssl.protocol]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[sslv2]]>
    </value>
</RuleCondition>
```

client.ssl.session.timeout

(Integer) Client SSL session timeout. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match client-side ssl session lifetime reaching 60 seconds

```
<RuleCondition>
    <subject>
    <![CDATA[client.ssl.session.timeout]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[60000]]>
    </value>
</RuleCondition>
```

client.tcp.port

(Integer) Used tcp port of the client request. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match client source-port 45678

```
<RuleCondition>
    <subject>
    <![CDATA[client.tcp.port]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[45678]]>
    </value>
</RuleCondition>
```

server.ip.address

(CIDR) IP address of the server. MATCH and NOT.MATCH accept regexp values.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH

Example

Match server ip address 172.26.10.123

```
<RuleCondition>
    <subject>
    <![CDATA[client.tcp.port]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[172.26.10.123]]>
    </value>
</RuleCondition>
```

server.ip.protocol

(List) IP protocol used by the server

EQUAL, NOT.EQUAL

Example

Match any IPV4 server address

```
<RuleCondition>
    <subject>
    <![CDATA[server.ip.protocol]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[IPV4]]>
    </value>
</RuleCondition>
```

server.ssl.protocol

(List) SSL protocol used by the server

EQUAL, NOT.EQUAL

Example

Match sslv2 protocol on server-side

```
<RuleCondition>
    <subject>
    <![CDATA[server.ssl.protocol]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[sslv2]]>
    </value>
</RuleCondition>
```

server.ssl.cipher

(String) SSL cipher used by the server SSL connection. MATCH and NOT.MATCH accept regexp values.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH

Example

Match server ssl cipher ECDHE-ECDSA-AES256-GCM-SHA384

```
<RuleCondition>
    <subject>
    <![CDATA[server.ssl.cipher]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[ECDHE-ECDSA-AES256-GCM-SHA384]]>
    </value>
</RuleCondition>
```

server.ssl.session.timeout

(Integer) Server SSL session timeout. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, IN-RANGE, NOT.IN-RANGE, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL

Example

Match server-side ssl session lifetime reaching 60 seconds

```
<RuleCondition>
    <subject>
    <![CDATA[server.ssl.session.timeout]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[60000]]>
    </value>
</RuleCondition>
```

server.tcp.port

(Integer) TCP port of the server. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, IN-RANGE, NOT.IN-RANGE, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL

Example

Match server tcp port 80

```
<RuleCondition>
    <subject>
    <![CDATA[server.tcp.port]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[80]]>
    </value>
</RuleCondition>
```

request.body.charset

(String) Charset defined in Content-Type request header. MATCH and NOT.MATCH accept regexp values.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH

Example

Match request body charset UTF-8

```
<RuleCondition>
    <subject>
    <![CDATA[request.body.charset]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[UTF-8]]>
    </value>
</RuleCondition>
```

request.body.length

(Integer) Request body length. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request body exceeding 50 bytes

```
<RuleCondition>
    <subject>
    <![CDATA[request.body.length]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[50]]>
    </value>
</RuleCondition>
```

request.body.parameter.specialkey

(String) Request body parameter key value. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request body parameter foo greater than 1000

```
<RuleCondition>
    <subject>
    <![CDATA[request.body.parameter]]>
    </subject>
    <key>
    <![foo]]></key>
   <operator>Equal</operator>
   <value>
    <![CDATA[1000]]>
    </value>
</RuleCondition>
```

request.body.parameter.count

(Integer) Count of request body parameters. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request body with more than 10 parameters

```
<RuleCondition>
    <subject>
    <![CDATA[request.body.parameter.count]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![CDATA[1000]]>
    </value>
</RuleCondition>
```

request.body.parameter.name

(String) Collection of request body parameters name. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request body parameter name foo

```
<RuleCondition>
    <subject>
    <![CDATA[request.body.parameter.name]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[foo]]>
    </value>
</RuleCondition>
```

request.body.parameter.name.length

(Integer) Length of the longest request body parameter name. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request with parameter name exceeding 50 bytes

```
<RuleCondition>
    <subject>
    <![CDATA[request.body.parameter.name.length]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![CDATA[50]]>
    </value>
</RuleCondition>
```

request.body.parameter.value

(String) Collection of request body parameters. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request parameter value foo

```
 <RuleCondition>
    <subject>
    <![CDATA[request.body.parameter.value]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[foo]]>
    </value>
</RuleCondition>
```

request.body.parameter.value.length

(Integer) Length of the longest request body parameter value. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request with parameter value exceeding 50 bytes

```
<RuleCondition>
    <subject>
    <![CDATA[request.body.parameter.value.length]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![CDATA[50]]>
    </value>
</RuleCondition>
```

request.body.type

(String) Content-Type of request header. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request body type application/json

```
<RuleCondition>
    <subject>
    <![CDATA[request.body.type]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[application/json]]>
    </value>
</RuleCondition>
```

request.duration

(Integer) Request duration. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request duration exceeding 1000 ms

```
<RuleCondition>
    <subject>
    <![CDATA[request.duration]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![CDATA[1000]]>
    </value>
</RuleCondition>
```

request.header.specialkey

(String) Name of the request header to access. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request header X-Forwarded-For: 8.8.8.8

```
<RuleCondition>
    <subject>
    <![CDATA[request.header]]>
    </subject>
    <key>
    <![CDATA[X-Forwarded-For]]></key>
   <operator>EQUAL</operator>
   <value>
    <![CDATA[8.8.8.8]]>
    </value>
</RuleCondition>
```

request.header.content-length

(Integer) Request header content length. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request content-length header greater than 500 bytes

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.content-length]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![CDATA[500]]>
    </value>
</RuleCondition>
```

request.header.content-type

(String) Request header content type. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match requests with application/json content

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.content-type]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[application/json]]>
    </value>
</RuleCondition>
```

request.header.cookie.specialkey

(String) Name of the cookie to access. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request cookie JSESSID=1234

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.cookie]]>
    </subject>
    <key>
    <![CDATA[JSESSID]]></key>
   <operator>EQUAL</operator>
   <value>
    <![CDATA[1234]]>
    </value>
</RuleCondition>
```

request.header.cookie.count

(Integer) Count of request cookies. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request cookie count greater than 5

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.cookie.count]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![CDATA[5]]>
    </value>
</RuleCondition>
```

request.header.cookie.name

(String) Name of the cookie to access. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request cookie JSESSID

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.cookie.name]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[JSESSID]]>
    </value>
</RuleCondition>
```

request.header.cookie.value

(String) Collection of request cookie values. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request cookie value 1234

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.cookie.value]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[1234]]>
    </value>
</RuleCondition>
```

request.header.line.count

(Integer) Count of request headers. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request header count not greater than 32

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.line.count]]>
    </subject>
    <operator>NOT.GREATER</operator>
    <value>
    <![CDATA[32]]>
    </value>
</RuleCondition>
```

request.header.line.length

(Integer) Longest request header length. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request header length not greater than 64 bytes

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.line.length]]>
    </subject>
    <operator>NOT.GREATER</operator>
    <value>
    <![CDATA[64]]>
    </value>
</RuleCondition>
```

request.header.name

(String) Collection of request header names. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request header X-Forwarded-For

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.name]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[X-Forwarded-For]]>
    </value>
</RuleCondition>
```

request.header.referer

(String) Request referer header value. MATCH and NOT.MATCH accept regexp values.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH

Example

Match request header referer
https://www.domain.com/flow/page?cart=abc&transaction=payment

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.referer]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[https://www.domain.com/flow/page?cart=abc&transaction=pay
ment]]>
    </value>
</RuleCondition>
```

request.header.user-agent

(String) Request User-Agent header value. MATCH and NOT.MATCH accept regexp values.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH

Example

Match user-agent curl

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.user-agent]]>
    </subject>
    <operator>MATCH</operator>
    <value>
    <![CDATA[^.*curl.*$]]>
    </value>
</RuleCondition>
```

request.header.value

(String) Collection of request header values. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request header value foo

```
<RuleCondition>
    <subject>
    <![CDATA[request.header.value]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[foo]]>
    </value>
</RuleCondition>
```

request.method

(List) Request method used

EQUAL, NOT.EQUAL

Example

Match request's HTTP method TRACE

```
<RuleCondition>
```

```
    <subject>
    <![CDATA[request.method]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[TRACE]]>
    </value>
</RuleCondition>
```

request.path

(String) Request URI path. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request path /admin/

```
<RuleCondition>
    <subject>
    <![CDATA[request.path]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[/admin/]]>
    </value>
</RuleCondition>
```

request.path.extension

(String) Request URI path extension. MATCH and NOT.MATCH accept regexp values.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH

Example

Match request file extension pdf

```
<RuleCondition>
    <subject>
    <![CDATA[request.path.extension]]>
```

```
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[pdf]]>
    </value>
</RuleCondition>
```

request.path.length

(Integer) Length of the request URI path. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request's path length not exceeding 1000 bytes

```
<RuleCondition>
    <subject>
    <![CDATA[request.path.extension]]>
    </subject>
    <operator>NOT.GREATER</operator>
    <value>
    <![CDATA[1000]]>
    </value>
</RuleCondition>
```

request.protocol

(List) Request protocol used

EQUAL, NOT.EQUAL

Example

Match request's HTTP protocol version 1.1

```
<RuleCondition>
    <subject>
    <![CDATA[request.protocol]]>
    </subject>
    <operator>EQUAL</operator>
```

```
    <value>
    <![CDATA[HTTP/1.1]]>
    </value>
</RuleCondition>
```

request.query-string

(String) Request URI query string. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request's query-string ?redirect-url=foo&user=admin

```
<RuleCondition>
    <subject>
    <![CDATA[request.query-string]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[redirect-url=foo&user=admin]]>
    </value>
</RuleCondition>
```

request.query-string.length

(Integer) Length of the request URI query string. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request query-string length beyond 1000

```
<RuleCondition>
    <subject>
    <![CDATA[request.query-string.length]]>
    </subject>
    <operator>GREATER</operator>
```

```
    <value>
    <![CDATA[1000]]>
    </value>
</RuleCondition>
```

request.query-string.parameter.specialkey

(String) Name of the query string parameter to access. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request query-string parameter foo with value equal 1000

```
<RuleCondition>
    <subject>
    <![CDATA[request.query-string.parameter]]>
    </subject>
    <key>
    <![CDATA[foo]]></key>
   <operator>EQUAL</operator>
   <value>
    <![CDATA[1000]]>
    </value>
</RuleCondition>
```

request.query-string.parameter.name

(String) Collection of request URI parameters. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request query-string parameter name foo

```
<RuleCondition>
    <subject>
    <![CDATA[request.query-string.parameter.name]]>
```

```
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[foo]]>
    </value>
</RuleCondition>
```

request.query-string.parameter.name.length

(Integer) Length of longest request URI parameter name. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request query-string parameter name greater than 10 chars

```
<RuleCondition>
    <subject>
    <![CDATA[request.query-string.parameter.name.length]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![CDATA[10]]>
    </value>
</RuleCondition>
```

request.query-string.parameter.count

(Integer) Count of request URI parameters. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request query-string with more than 10 parameters

```
<RuleCondition>
    <subject>
```

```
    <![CDATA[request.query-string.parameter.count]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![CDATA[10]]>
    </value>
</RuleCondition>
```

request.query-string.parameter.value

(String) The request URI parameters. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request parameter value foo

```
<RuleCondition>
    <subject>
    <![CDATA[request.query-string.parameter.value]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[foo]]>
    </value>
</RuleCondition>
```

request.query-string.parameter.value.length

(Integer) Length of longest request URI parameter value. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request parameter value length 64

```
<RuleCondition>
    <subject>
```

```
    <![CDATA[request.query-string.parameter.value.length]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[64]]>
    </value>
</RuleCondition>
```

request.url

(String) Client request URL. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match request url that is not /foo/bar/resource?param=value

```
<RuleCondition>
    <subject>
    <![CDATA[request.url]]>
    </subject>
    <operator>NOT.EQUAL</operator>
    <value>
    <![CDATA[/foo/bar/resource?param=value]]>
    </value>
</RuleCondition>
```

request.url.length

(Integer) Length of request URL. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match request url length not exceeding 1000 bytes

```
<RuleCondition>
    <subject>
```

```
    <![CDATA[request.url.length]]>
    </subject>
    <operator>NOT.GREATER</operator>
    <value>
    <![CDATA[1000]]>
    </value>
</RuleCondition>
```

response.code

(Integer) HTTP code of the response. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match response code 500

```
<RuleCondition>
    <subject>
    <![CDATA[response.code]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[500]]>
    </value>
</RuleCondition>
```

response.date

(Date) Date of the response. BETWEEN and NOT.BETWEEN accept dateRange values.

EQUAL, NOT.EQUAL, BETWEEN, NOT.BETWEEN

Example

Match response date between 02-04-2017 and 02-08-2017

```
<RuleCondition>
    <subject>
    <![CDATA[response.date]]>
```

```
    </subject>
    <operator>BETWEEN</operator>
    <value>
    <![CDATA[02-04-2017,02-08-2017]]>
    </value>
</RuleCondition>
```

response.duration

(Integer) Duration of the response generation. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match response duration exceeding 1000 ms

```
<RuleCondition>
    <subject>
    <![CDATA[response.duration]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![CDATA[1000]]>
    </value>
</RuleCondition>
```

response.header.specialkey

(String) Access to a specific response header. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match response header X-Forwarded-For: 8.8.8.8

```
<RuleCondition>
    <subject>
    <![CDATA[response.header]]>
```

```
    </subject>
    <key>
    <![CDATA[X-Forwarded-For]]></key>
   <operator>EQUAL</operator>
   <value>
    <![CDATA[8.8.8.8]]>
    </value>
</RuleCondition>
```

response.header.content-length

(Integer) Content length of the response. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match response content-length header greater than 500 bytes

```
<RuleCondition>
    <subject>
    <![CDATA[response.header.content-length]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![CDATA[500]]>
    </value>
</RuleCondition>
```

response.header.content-type

(String) Content type of the response. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match responses with application/json content

```
<RuleCondition>
```

```
    <subject>
    <![CDATA[response.header.content-type]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![CDATA[application/json]]>
    </value>
</RuleCondition>
```

response.header.cookie.specialkey

(String) Access to the value of a specific response cookie. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match response cookie JSESSID=1234

```
<RuleCondition>
    <subject>
    <![CDATA[response.header.cookie]]>
    </subject>
    <key>
    <![CDATA[JSESSID]]></key>
   <operator>EQUAL</operator>
   <value>
    <![CDATA[1234]]>
    </value>
</RuleCondition>
```

response.header.cookie.name

(String) Collection of response header cookies names. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match response cookie JSESSID

```
<RuleCondition>
    <subject>
    <![CDATA[response.header.cookie.name]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![JSESSID]]>
    </value>
</RuleCondition>
```

response.header.cookie.value

(String) Collection of response header cookies values. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match response cookie value 1234

```
<RuleCondition>
    <subject>
    <![CDATA[response.header.cookie.value]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![1234]]>
    </value>
</RuleCondition>
```

response.header.line.length

(Integer) Length of the response line. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match response header length not greater than 64 bytes

```
<RuleCondition>
    <subject>
    <![CDATA[response.header.line.length]]>
    </subject>
    <operator>NOT.GREATER</operator>
    <value>
    <![64]]>
    </value>
</RuleCondition>
```

response.header.name

(String) Collection of response header names. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match response header X-Forwarded-For

```
<RuleCondition>
    <subject>
    <![CDATA[response.header.name]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![X-Forwarded-For]]>
    </value>
</RuleCondition>
```

response.header.value

(String) Collection of response header values. MATCH and NOT.MATCH accept regexp values. DETECT acts on a keyword and takes as value the qid or the tag of the detection rule that must be allowed on this location.

EQUAL, NOT.EQUAL, MATCH, NOT.MATCH, DETECT

Example

Match response header value foo

```
<RuleCondition>
    <subject>
    <![CDATA[response.header.value]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![foo]]>
    </value>
</RuleCondition>
```

response.time

(Time) Time of the response. BETWEEN and NOT.BETWEEN accept timeRange values.

EQUAL, NOT.EQUAL, BETWEEN, NOT.BETWEEN

Example

Match response time between 08:00 and 19:00

```
<RuleCondition>
    <subject>
    <![CDATA[response.time]]>
    </subject>
    <operator>BETWEEN</operator>
    <value>
    <![08:00,19:00]]>
    </value>
</RuleCondition>
```

response.protocol

(List) Protocol of the response

EQUAL, NOT.EQUAL

Example

Match response's HTTP protocol version 1.1

```
<RuleCondition>
    <subject>
    <![CDATA[response.protocol]]>
    </subject>
```

```
    <operator>EQUAL</operator>
    <value>
    <![HTTP/1.1]]>
    </value>
</RuleCondition>
```

transaction.date

(Date) Date of transaction. BETWEEN and NOT.BETWEEN accept dateRange values.

EQUAL, NOT.EQUAL, BETWEEN, NOT.BETWEEN

Example

Match transaction date between 02-04-2017 and 02-08-2017

```
<RuleCondition>
    <subject>
    <![CDATA[transaction.date]]>
    </subject>
    <operator>BETWEEN</operator>
    <value>
    <![02-04-2017,02-08-2017]>
    </value>
</RuleCondition>
```

transaction.day

(List) Day of transaction

EQUAL, NOT.EQUAL

Example

Match transaction day Sunday

```
<RuleCondition>
    <subject>
    <![CDATA[transaction.day]]>
    </subject>
    <operator>EQUAL</operator>
    <value>
    <![Sunday]>
    </value>
```

```
</RuleCondition>
```

transaction.duration

(Integer) Duration of the full transaction. IN-RANGE and NOT.IN-RANGE accept intRange values.

EQUAL, NOT.EQUAL, GREATER, NOT.GREATER, GREATER-EQUAL, NOT.GREATER-EQUAL, LOWER, NOT.LOWER, LOWER-EQUAL, NOT.LOWER-EQUAL, IN-RANGE, NOT.IN-RANGE

Example

Match transation duration exceeding 1000 ms

```
<RuleCondition>
    <subject>
    <![CDATA[transaction.duration]]>
    </subject>
    <operator>GREATER</operator>
    <value>
    <![1000]>
    </value>
</RuleCondition>
```

transaction.time

(Time) Time of transaction. BETWEEN and NOT.BETWEEN accept timeRange values.

EQUAL, NOT.EQUAL, BETWEEN, NOT.BETWEEN.

Example

Match transaction time between 08:00 and 19:00

```
<RuleCondition>
    <subject>
    <![CDATA[transaction.date]]>
    </subject>
    <operator>BETWEEN</operator>
    <value>
    <![08:00,19:00]>
    </value>
</RuleCondition>
```

Clusters API

Current cluster count

/qps/rest/2.0/count/waf/cluster

[GET]

Returns the total number of WAF clusters in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Manage WAFs" permission, and clusters licensed for WAF and within the user's scope.

Input Parameters

No input elements are available.

Sample

API request

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

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/
xsd/2.0/waf/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>52</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/cluster.xsd

Get details on a cluster

/qps/rest/2.0/get/waf/cluster/<id>

[GET]

Returns details about a specific WAF cluster in the user's account. Want to find a cluster ID to use as input? See Search clusters.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Manage WAFs" permission, and clusters licensed for WAF and within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the cluster ID of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/get/waf/cluster/25401"
```

Response

```
<?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
/waf/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Cluster>
      <id>25401</id>
      <uuid>7de38215-a9e3-4855-8566-d9b369a92756</uuid>
      <name><![CDATA[Site1]]></name>
      <description><![CDATA[Manage all Appliances from
Montlegun]]></description>
      <errorResponse>
```

```
      <redirect>
        <url>https://www.domain.com/resturl.html</url>
        <status>302</status>
      </redirect>
    </errorResponse>
    <trustedIPs>
      <string>1.2.3.5</string>
      <string>1.2.3.6/23</string>
    </trustedIPs>
    <updateSchedule>
      <enabled>true</enabled>
      <weekDays>MON,THU</weekDays>
      <startTime>23</startTime>
      <timezone>
          <code>Asia/Aden</code>
          <offset>+03:00</offset>
      </timezone>
      <freezeEndDate>2018-02-28</freezeEndDate>
      <nextUpgradeDate>2018-03-01T23:00:00Z</nextUpgradeDate>
    </updateSchedule>
    <owner>
      <id>3989443</id>
      <username>john_doe</username>
      <firstname>John</firstname>
      <lastname>Doe</lastname>
    </owner>
    <created>2017-07-25T09:35:01Z</created>
    <createdBy>
      <id>3989443</id>
      <username>john_doe</username>
      <firstname>John</firstname>
      <lastname>Doe</lastname>
    </createdBy>
    <updated>2017-07-25T10:11:13Z</updated>
    <updatedBy>
      <id>3989442</id>
      <username>john_doe</username>
      <firstname>John</firstname>
      <lastname>Doe</lastname>
    </updatedBy>
    <token><![CDATA[C190F3A8-08FA-46DD-BB16-EB14ECC47606]]></token>
    <syncDate>2017-07-25T10:22:30Z</syncDate>
    <status>DEGRADED</status>
    <deploymentStatus>SUCCESS</deploymentStatus>
    <deployed>2017-10-23T09:22:34Z</deployed>
```

```
      <appliances>
        <Appliance>
          <id>15802</id>
          <uuid>580c5d0f-b80b-452d-a345-0ee1526fe88f</uuid>
          <name><![CDATA[580C5D0F-B80B-452D-A345-
0EE1526FE88F]]></name>
        </Appliance>
        <Appliance>
          <id>15803</id>
          <uuid>6c5844e9-86f5-418e-90d5-b0201123bf13</uuid>
          <name><![CDATA[6C5844E9-86F5-418E-90D5-
B0201123BF13]]></name>
        </Appliance>
      </appliances>
      <tags>
        <Tag>
          <id>10256057</id>
          <name><![CDATA[KIDS]]></name>
        </Tag>
      </tags>
    </Cluster>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/cluster.xsd

Search clusters

/qps/rest/2.0/search/waf/cluster

[POST]

Finds WAF clusters in the user's account matching the search criteria.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Manage WAFs" permission, and clusters 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. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Supported filter operators

Parameter	Description
id (Long)	Cluster identifier on Qualys Cloud Platform.
uuid (UUID)	Cluster identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the cluster as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the WAF cluster as defined by a user. Valid action: Update

owner (Text)	The user for Qualys Cloud Platform who owns this cluster.
owner.id (Long)	The user ID of the WAF cluster owner.
owner.username (Text)	The user name of the WAF cluster owner.
owner.firstname (Text)	The first name of the WAF cluster owner.
owner.lastname (Text)	The last name of the WAF cluster owner.
created (Date)	The date/time when the WAF cluster was created.
createdBy.id (Long)	The user ID who created the WAF cluster.
createdBy.username (Text)	The user name who created the WAF cluster.
createdBy.firstname (Text)	The first name of the user who created the WAF cluster.
createdBy.lastname (Text)	The last name of the user who created the WAF cluster.
updated (Date)	The date/time when the WAF cluster was last updated.
updatedBy.id (Long)	The user ID who last updated the WAF cluster.
updatedBy.username (Text)	The user name who last updated the WAF cluster.
updatedBy.firstname (Text)	The first name of the user who

	updated the WAF cluster.
updatedBy.lastname (Text)	The last name of the user who updated the WAF cluster.
token (Text)	The registration token for the WAF cluster.
syncDate (Date)	The last synchronization date of assets (between Qualys Cloud platform database and WAF database).
status (Text)	Status information: NO_SENSORS (No appliances are connected to the WAF cluster) INACTIVE (All appliances connected to the WAF cluster are up and ready for service, but the configuration is yet to be deployed) DEGRADED (At least one appliance is connected to the WAF cluster, but other appliances are disconnected) ACTIVE (All appliances connected to the WAF cluster are up and ready for service) ERROR (All appliances connected to the WAF cluster are disconnected)
deploymentStatus (Text)	Deployment Status information: SUCCESS (Configuration

	deployment was successful on all appliances in a cluster) PENDING_DEPLOY (Configuration deployment has been requested but not yet started) FAILURE (Configuration deployment has failed on all appliances in a cluster) PARTIAL (Configuration deployment was successful on some appliances but failed on other appliances in a cluster) IN_PROGRESS (Configuration deployment is in progress on all appliances in a cluster) UNUSED (Web application is not deployed on any WAF cluster, or the Web application is deployed on a cluster having no appliances registered to it)
deployed (Date)	The last deployment date of the cluster.
errorResponse.action (Text)	Error Response behavior (block, redirect, custom response) when a request is not routable.
errorResponse.redirect.url (Text)	Empty or a valid URL used to redirect web client when request cannot be routed to any known web application.
errorResponse.redirect.status (Long)	A valid redirection HTTP code (301/302) used to redirect web

	client (see redirect.url)
errorResponse.customPage.id (Long)	The ID of the custom response page to assign to the cluster.
errorResponse.customPage.uuid (UUID)	The UUID of the custom response page to assign to the cluster.
errorResponse.customPage.name (Text)	The name of the custom response page to assign to the cluster.
webApps.webApp.id (Long)	The ID of the Web Application this WAF cluster is associated with.
webApps.webApp.uuid (UUID)	The UUID of the Web Application this WAF cluster is associated with.
webApps.webApp.name (Text)	The name of the Web Application this WAF cluster is associated with.
trustedIPs.string (Text)	Provide the IP address/range/network of trusted origin proxies or load balancers configured in full-proxy mode. Enter an IP address between 1.0.0.0 - 223.255.255.254 excluding 127.x.x.x. Optionally, you can provide a range for the IP address between 1 - 32. For example, 1.2.3.6/23 If the request is not from a trusted source the X-Forwarded-For header values

	are automatically discarded. If you do not provide IP addresses for trusted origin proxies or load balancers, then IP addresses as per RFC1918 are trusted.
updateSchedule.enabled (Boolean)	Schedule Auto-Update enabled or disabled. By default this is enabled.
tags.id (Long)	The ID of a tag associated with the WAF cluster.
tags.name (Text)	The name, defined by a user, of a tag associated with the WAF cluster.

Sample

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/cluster"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="EQUALS">Site1</Criteria>
    <Criteria field="errorResponse.redirect.url"
operator="CONTAINS">site1</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <Cluster>
      <id>25401</id>
      <uuid>7de38215-a9e3-4855-8566-d9b369a92756</uuid>
      <name><![CDATA[Site1]]></name>
      <description><![CDATA[Manage all Appliances from
Site1]]></description>
      <errorResponse>
        <redirect>
          <url>https://www.domain.com/resturl.html</url>
          <status>302</status>
        </redirect>
      </errorResponse>
      <trustedIPs>
        <string>1.2.3.5</string>
        <string>1.2.3.6/23</string>
      </trustedIPs>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,THU</weekDays>
        <startTime>23</startTime>
        <timezone>
            <code>Asia/Aden</code>
            <offset>+03:00</offset>
        </timezone>
        <freezeEndDate>2018-02-28</freezeEndDate>
        <nextUpgradeDate>2018-03-01T23:00:00Z</nextUpgradeDate>
      </updateSchedule>
      <owner>
        <id>3989443</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </owner>
      <created>2017-07-25T09:35:01Z</created>
      <createdBy>
        <id>3989443</id>
        <username>john_doe</username>
```

```
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </createdBy>
      <updated>2017-07-25T10:11:13Z</updated>
      <updatedBy>
        <id>3989442</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </updatedBy>
      <token><![CDATA[C190F3A8-08FA-46DD-BB16-EB14ECC47606]]></token>
      <syncDate>2017-07-25T10:52:30Z</syncDate>
      <status>DEGRADED</status>
      <deploymentStatus>SUCCESS</deploymentStatus>
      <deployed>2017-10-23T09:22:34Z</deployed>
      <appliances>
        <Appliance>
          <id>15802</id>
          <uuid>580c5d0f-b80b-452d-a345-0ee1526fe88f</uuid>
          <name><![CDATA[580C5D0F-B80B-452D-A345-
0EE1526FE88F]]></name>
        </Appliance>
        <Appliance>
          <id>15803</id>
          <uuid>6c5844e9-86f5-418e-90d5-b0201123bf13</uuid>
          <name><![CDATA[6C5844E9-86F5-418E-90D5-
B0201123BF13]]></name>
        </Appliance>
      </appliances>
      <tags>
        <Tag>
          <id>10256057</id>
          <name><![CDATA[KIDS]]></name>
        </Tag>
      </tags>
    </Cluster>
  </data>
</ServiceResponse>
      <tags>
        <Tag>
          <id>7530430</id>
          <name><![CDATA[Cloud Agent]]></name>
        </Tag>
      </tags>
      <description><![CDATA[Cluster description added]]></description>
```

```
      <redirectUrl>http://mydomain.com</redirectUrl>
      <httpErrorCode>302</httpErrorCode>
    </Cluster>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/cluster.xsd

Search clusters

/qps/rest/2.0/create/waf/cluster

[POST]

Create a WAF cluster.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Manage WAFs" permission, and clusters 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.

Supported filter operators

Parameter	Description
name (Text)	(Required) The name of the cluster as defined by a user. This is unique in subscription. Valid action: Update
updateSchedule.enabled (Boolean) - Required only when using the updateSchedule element.	(Required) Schedule Auto-Update enabled or disabled. By default this is enabled.
description (Text)	The description of the WAF cluster as defined by a user. Valid action: Update
errorResponse.action (Text)	Error Response behavior (block, redirect, custom response) when

	a request is not routable.
errorResponse.redirect.url (Text)	Empty or a valid URL used to redirect web client when request cannot be routed to any known web application.
errorResponse.redirect.status (Long)	A valid redirection HTTP code (301/302) used to redirect web client (see redirect.url)
errorResponse.customPage.id (Long)	The ID of the custom response page to assign to the cluster.
trustedIPs.string (Text)	Provide the IP address/range/network of trusted origin proxies or load balancers configured in full-proxy mode. Enter an IP address between 1.0.0.0 - 223.255.255.254 excluding 127.x.x.x. Optionally, you can provide a range for the IP address between 1 - 32. For example, 1.2.3.6/23 If the request is not from a trusted source the X-Forwarded-For header values are automatically discarded. If you do not provide IP addresses for trusted origin proxies or load balancers, then IP addresses as per RFC1918 are trusted.
updateSchedule.weekDays (Text)	Comma separated list of days the auto-update is allowed. By default enabled for all days: MON,TUE,WED,THU,FRI,SAT,SUN.

updateSchedule.startTime (Integer)	Hour of the day the auto-update may occur. Default is 0. Specify the hour in 24 hour format.
updateSchedule.timezone.code (Text)	The timezone. For example, Pacific Standard Time. Default is UTC.
updateSchedule.timezone.offset (Text)	The timezone offset (hours and minutes). Default is +00:00.
updateSchedule.freezeEndDate (Date)	If set the auto-update process will not be executed before the specified date (e.g. 2018-03-01). Disabled by default.
tags	Provide a list of existing tags to be assigned to this cluster. Multiple tags can be provided in the form of a comma separated list.
tags.id (Long)	The ID of a tag associated with the WAF cluster.
tags.name (Text)	The name, defined by a user, of a tag associated with the WAF cluster.

Sample

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/cluster"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <data>
    <Cluster>
      <name>Site1</name>
      <description>Manage all Appliances from Site1</description>
      <errorResponse>
        <redirect>
          <url>https://www.domain.com/resturl.html</url>
          <status>302</status>
        </redirect>
      </errorResponse>
      <trustedIPs>
        <string>1.2.3.5</string>
        <string>1.2.3.6/23</string>
      </trustedIPs>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,THU</weekDays>
        <startTime>23</startTime>
        <timezone>
            <code>Asia/Aden</code>
            <offset>+03:00</offset>
        </timezone>
        <freezeEndDate>2018-02-28</freezeEndDate>
      </updateSchedule>
    </Cluster>
  </data>
</ServiceRequest>
```

Response

```
<?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
/waf/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Cluster>
      <id>25401</id>
      <uuid>7de38215-a9e3-4855-8566-d9b369a92756</uuid>
      <name><![CDATA[Site1]]></name>
      <description><![CDATA[Manage all Appliances from
Site1]]></description>
```

```
      <errorResponse>
        <redirect>
          <url>https://www.domain.com/resturl.html</url>
          <status>302</status>
        </redirect>
      </errorResponse>
      <trustedIPs>
        <string>1.2.3.5</string>
        <string>1.2.3.6/23</string>
      </trustedIPs>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,THU</weekDays>
        <startTime>23</startTime>
        <timezone>
            <code>Asia/Aden</code>
            <offset>+03:00</offset>
        </timezone>
        <freezeEndDate>2018-02-28</freezeEndDate>
        <nextUpgradeDate>2018-03-01T23:00:00Z</nextUpgradeDate>
      </updateSchedule>
      <owner>
        <id>3989443</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </owner>
      <created>2017-07-25T09:35:01Z</created>
      <createdBy>
        <id>3989443</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </createdBy>
      <updated>2017-07-25T09:35:01Z</updated>
      <updatedBy>
        <id>3989443</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </updatedBy>
      <token><![CDATA[C190F3A8-08FA-46DD-BB16-EB14ECC47606]]></token>
      <status>NO_SENSORS</status>
    </Cluster>
  </data>
```

```
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/cluster.xsd

Update cluster

/qps/rest/2.0/update/waf/cluster/<id>

[POST]

Update a WAF cluster in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Manage WAFs" permission, and clusters licensed for WAF and within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the cluster you want to update.

Optional input elements are listed below. The associated data type for each element appears in parentheses.

Supported filter operators

Parameter	Description
name (Text)	The name of the cluster as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the WAF cluster as defined by a user. Valid action: Update
errorResponse.action (Text)	Error Response behavior (block, redirect, custom response) when a request is not routable.

errorResponse.redirect.url (Text)	Empty or a valid URL used to redirect web client when request cannot be routed to any known web application.
errorResponse.redirect.status (Long)	A valid redirection HTTP code (301/302) used to redirect web client (see redirect.url)
errorResponse.customPage.id (Long)	The ID of the custom response page to assign to the cluster.
errorResponse.customPage.uuid (UUID)	The UUID of the custom response page to assign to the cluster.
errorResponse.customPage.name (Text)	The name of the custom response page to assign to the cluster.
trustedIPs.string (Text)	Provide the IP address/range/network of trusted origin proxies or load balancers configured in full-proxy mode. Enter an IP address between 1.0.0.0 - 223.255.255.254 excluding 127.x.x.x. Optionally, you can provide a range for the IP address between 1 - 32. For example, 1.2.3.6/23 If the request is not from a trusted source the X-Forwarded-For header values are automatically discarded. If you do not provide IP addresses for trusted origin proxies or load balancers, then IP addresses as per RFC1918 are trusted.
updateSchedule.enabled (Boolean)	Schedule Auto-Update enabled

	or disabled. By default this is enabled.
updateSchedule.weekDays (Text)	Comma separated list of days the auto-update is allowed. By default enabled for all days: MON,TUE,WED,THU,FRI,SAT,SUN.
updateSchedule.startTime (Integer)	Hour of the day the auto-update may occur. Default is 0. Specify the hour in 24 hour format.
updateSchedule.timezone.code (Text)	The timezone. For example, Pacific Standard Time. Default is UTC.
updateSchedule.timezone.offset (Text)	The timezone offset (hours and minutes). Default is +00:00.
updateSchedule.freezeEndDate (Date)	If set the auto-update process will not be executed before the specified date (e.g. 2018-03-01). Disabled by default.
tags	Provide a list of existing tags to be assigned to this cluster. Multiple tags can be provided in the form of a comma separated list.

Sample

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/cluster/25401"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <data>
    <Cluster>
      <name>Site1 Updated</name>
      <tags><Tag><name>ABC</name></Tag></tags>
    </Cluster>
  </data>
</ServiceRequest>
```

Response

```
<?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
/waf/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Cluster>
      <id>25401</id>
      <uuid>7de38215-a9e3-4855-8566-d9b369a92756</uuid>
      <name><![CDATA[Site1 Updated]]></name>
      <description><![CDATA[Manage all Appliances from
Site1]]></description>
      <errorResponse>
        <redirect>
          <url>https://www.domain.com/resturl.html</url>
          <status>302</status>
        </redirect>
      </errorResponse>
      <trustedIPs>
        <string>1.2.3.5</string>
        <string>1.2.3.6/23</string>
      </trustedIPs>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,THU</weekDays>
        <startTime>23</startTime>
        <timezone>
            <code>Asia/Aden</code>
            <offset>+03:00</offset>
        </timezone>
        <freezeEndDate>2018-02-28</freezeEndDate>
```

```
        <nextUpgradeDate>2018-03-01T23:00:00Z</nextUpgradeDate>
      </updateSchedule>
      <owner>
        <id>3989443</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </owner>
      <created>2017-07-25T09:35:01Z</created>
      <createdBy>
        <id>3989443</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </createdBy>
      <updated>2017-07-25T12:22:44Z</updated>
      <updatedBy>
        <id>3989444</id>
        <username>john_doe</username>
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </updatedBy>
      <token><![CDATA[C190F3A8-08FA-46DD-BB16-EB14ECC47606]]></token>
      <syncDate>2017-07-25T12:22:30Z</syncDate>
      <status>DEGRADED</status>
      <appliances>
        <Appliance>
          <id>15802</id>
          <uuid>580c5d0f-b80b-452d-a345-0ee1526fe88f</uuid>
          <name><![CDATA[580C5D0F-B80B-452D-A345-
0EE1526FE88F]]></name>
        </Appliance>
        <Appliance>
          <id>15803</id>
          <uuid>6c5844e9-86f5-418e-90d5-b0201123bf13</uuid>
          <name><![CDATA[6C5844E9-86F5-418E-90D5-
B0201123BF13]]></name>
        </Appliance>
      </appliances>
      <tags>
        <Tag>
          <id>10256057</id>
          <name><![CDATA[ABC]]></name>
        </Tag>
      </tags>
```

```
    </Cluster>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/cluster.xsd

Update clusters (bulk)

/qps/rest/2.0/update/waf/cluster

[POST]

Update multiple WAF clusters in the user's account. You can update all fields except tag ID and tag name.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Manage WAFs" permission, and clusters licensed for WAF and within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search clusters.

Allowed input elements for bulk update are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Supported filter operators

Parameter	Description
name (Text)	The name of the cluster as defined by a user. This is unique in subscription. Valid action: Update
description (Text)	The description of the WAF cluster as defined by a user. Valid action: Update
errorResponse.action (Text)	Error Response behavior (block, redirect, custom response) when

	a request is not routable.
errorResponse.redirect.url (Text)	Empty or a valid URL used to redirect web client when request cannot be routed to any known web application.
errorResponse.redirect.status (Long)	A valid redirection HTTP code (301/302) used to redirect web client (see redirect.url)
errorResponse.customPage.id (Long)	The ID of the custom response page to assign to the cluster.
errorResponse.customPage.uuid (UUID)	The UUID of the custom response page to assign to the cluster.
errorResponse.customPage.name (Text)	The name of the custom response page to assign to the cluster.
trustedIPs.string (Text)	Provide the IP address/range/network of trusted origin proxies or load balancers configured in full-proxy mode. Enter an IP address between 1.0.0.0 - 223.255.255.254 excluding 127.x.x.x. Optionally, you can provide a range for the IP address between 1 - 32. For example, 1.2.3.6/23 If the request is not from a trusted source the X-Forwarded-For header values are automatically discarded. If you do not provide IP addresses for trusted origin proxies or load balancers, then IP addresses as per RFC1918 are trusted.

updateSchedule.enabled (Boolean)	Schedule Auto-Update enabled or disabled. By default this is enabled.
updateSchedule.weekDays (Text)	Comma separated list of days the auto-update is allowed. By default enabled for all days: MON,TUE,WED,THU,FRI,SAT,SUN.
updateSchedule.startTime (Integer)	Hour of the day the auto-update may occur. Default is 0. Specify the hour in 24 hour format.
updateSchedule.timezone.code (Text)	The timezone. For example, Pacific Standard Time. Default is UTC.
updateSchedule.timezone.offset (Text)	The timezone offset (hours and minutes). Default is +00:00.
updateSchedule.freezeEndDate (Date)	If set the auto-update process will not be executed before the specified date (e.g. 2018-03-01). Disabled by default.
tags	Provide a list of existing tags to be assigned to this cluster. Multiple tags can be provided in the form of a comma separated list.

Sample

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/cluster"

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="createdBy.firstname"
operator="EQUALS">John</Criteria>
  </filters>
  <data>
    <Cluster>
      <tags><Tag><name>XYZ</name></Tag></tags>
    </Cluster>
  </data>
</ServiceRequest>
```

Response

```
<?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/waf/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Cluster>
      <id>25401</id>
      <uuid>7de38215-a9e3-4855-8566-d9b369a92756</uuid>
      <name><![CDATA[Montlegun Updated]]></name>
      <description><![CDATA[Manage all Appliances from
Montlegun]]></description>
      <errorResponse>
        <redirect>
          <url>https://www.domain.com/resturl.html</url>
          <status>302</status>
        </redirect>
      </errorResponse>
      <trustedIPs>
        <string>1.2.3.5</string>
        <string>1.2.3.6/23</string>
      </trustedIPs>
      <updateSchedule>
        <enabled>true</enabled>
        <weekDays>MON,THU</weekDays>
        <startTime>23</startTime>
        <timezone>
            <code>Asia/Aden</code>
```

```
            <offset>+03:00</offset>
          </timezone>
          <freezeEndDate>2018-02-28</freezeEndDate>
          <nextUpgradeDate>2018-03-01T23:00:00Z</nextUpgradeDate>
        </updateSchedule>
        <owner>
          <id>3989443</id>
          <username>john_doe</username>
          <firstname>John</firstname>
          <lastname>Doe</lastname>
        </owner>
        <created>2017-07-25T09:35:01Z</created>
        <createdBy>
          <id>3989443</id>
          <username>john_doe</username>
          <firstname>John</firstname>
          <lastname>Doe</lastname>
        </createdBy>
        <updated>2017-07-25T12:30:24Z</updated>
        <updatedBy>
          <id>3989444</id>
          <username>john_doe</username>
          <firstname>John</firstname>
          <lastname>Doe</lastname>
        </updatedBy>
        <token><![CDATA[C190F3A8-08FA-46DD-BB16-EB14ECC47606]]></token>
        <syncDate>2017-07-25T12:27:31Z</syncDate>
        <status>DEGRADED</status>
        <appliances>
          <Appliance>
            <id>15802</id>
            <uuid>580c5d0f-b80b-452d-a345-0ee1526fe88f</uuid>
            <name><![CDATA[580C5D0F-B80B-452D-A345-
0EE1526FE88F]]></name>
          </Appliance>
          <Appliance>
            <id>15803</id>
            <uuid>6c5844e9-86f5-418e-90d5-b0201123bf13</uuid>
            <name><![CDATA[6C5844E9-86F5-418E-90D5-
B0201123BF13]]></name>
          </Appliance>
        </appliances>
        <tags>
          <Tag>
            <id>10256059</id>
```

```
          <name><![CDATA[XYZ]]></name>
        </Tag>
      </tags>
    </Cluster>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/cluster.xsd

Delete cluster

/qps/rest/2.0/delete/waf/cluster/<id>

[POST]

Delete a WAF cluster in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Manage WAFs" permission, and clusters licensed for WAF and within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the cluster asset you want to delete.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "POST" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/delete/waf/cluster/122801"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/
xsd/2.0/waf/cluster.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Cluster>
      <id>122801</id>
    </Cluster>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/cluster.xsd

Delete clusters (bulk)

/qps/rest/2.0/delete/waf/cluster

[POST]

Delete multiple WAF clusters in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Manage WAFs" permission, and clusters licensed for WAF and within the user's scope.

Input Parameters

All elements for the search operation are supported. See Search clusters.

Sample

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/delete/waf/cluster"

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

Request POST data

```
<?xml version="1.0" ?>
<ServiceRequest>
  <filters>
    <Criteria field="tags.tag.name" operator="EQUALS">XYZ</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?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
/waf/cluster.xsd">
```

```
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Cluster>
      <id>25401</id>
    </Cluster>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/cluster.xsd

Appliances API

Current appliance count

/qps/rest/2.0/count/waf/appliance

[GET]

Returns the total number of WAF appliances in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and appliances licensed for WAF and within the user's scope.

Input Parameters

No input elements are available.

Sample

API request

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

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/
xsd/2.0/waf/appliance.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>4</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/appliance.xsd

Get details on appliance

/qps/rest/2.0/get/waf/appliance/:id

[GET]

Returns details about a specific WAF appliance in the user's account. Want to find an appliance ID to use as input? See Search appliances.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and appliances licensed for WAF and within the user's scope.

Input Parameters

The element "id" (Integer) is required, where "id" identifies the appliance ID of interest.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "GET" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/get/waf/appliance/15804"
```

Response

```
<?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
/waf/appliance.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Appliance>
      <id>15804</id>
      <uuid>9276b0ed-53e4-4c43-b27e-463851990ee3</uuid>
      <name><![CDATA[9276B0ED-53E4-4C43-B27E-463851990EE3]]></name>
      <hostname><![CDATA[30c1def71a9d]]></hostname>
      <lastPollDate>2017-07-25T13:13:15Z</lastPollDate>
      <applianceCreated>2017-07-25T12:51:03Z</applianceCreated>
```

```
      <applianceVersion><![CDATA[1.4.0]]></applianceVersion>
      <status>INACTIVE</status>
      <pollStatus>NOT_POLLING</pollStatus>
      <heartbeatGenerated>2017-07-25T13:13:11Z</heartbeatGenerated>
      <heartbeatProcessed>2017-07-25T13:13:12Z</heartbeatProcessed>
      <systemOs><![CDATA[Linux 30c1def71a9d 4.4.0-83-generic #106-
Ubuntu SMP Mon Jun 26 17:54:43 UTC 2017 x86_64]]></systemOs>
      <systemRam>12284530688</systemRam>
      <systemType><![CDATA[other]]></systemType>
      <systemCpusCount>1</systemCpusCount>
      <systemCpusCores>2</systemCpusCores>
      <systemCpusSpeed>2600.406</systemCpusSpeed>
      <systemCpusModel><![CDATA[Intel(R) Core(TM) i7-5600U CPU @
2.60GHz]]></systemCpusModel>
     <configRulesVersion><![CDATA[2017.7.14.269]]></configRulesVersion
>
      <configVersion><![CDATA[2017-07-
25T13:07:59.244Z]]></configVersion>
      <configGenerated>2017-07-25T13:07:59Z</configGenerated>
      <ip><![CDATA[172.17.0.2]]></ip>
      <cluster>
        <id>25603</id>
        <uuid>d817eb34-2320-4044-9c0c-8b0b2523a75a</uuid>
        <name><![CDATA[Montlegun]]></name>
      </cluster>
    </Appliance>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/appliance.xsd

Search appliances

/qps/rest/2.0/search/waf/appliance

[POST]

Finds WAF appliances in the user's account matching search criteria.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, and appliances 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. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Supported filter operators

Parameter	Description
id (Long)	WAF appliance identifier on Qualys Cloud Platform.
uuid (UUID)	WAF appliance identifier within the Qualys Cloud WAF Service.
name (Text)	The name of the waf appliance as defined by a user. This is unique in subscription. Valid action: Update
hostname (Text)	The hostname of the appliance (retrieved asynchronously).
lastPollDate (Date)	The last poll date of appliance. If the

	appliance does not poll for the commands, connectivity error or failure status is detected.
applianceCreated (Long)	Appliance creation date in the Qualys Cloud WAF Service.
applianceVersion (Text)	Software version of Qualys WAF Service running on the appliance.
status (Text)	The appliance status information: ACTIVE, INACTIVE, DEPROVISIONING (appliance is currently cleaning configuration before shutdown for appliance deletion), DISABLED (appliance is removed from any cluster and is been destroyed. No appliance records should be retrieved with that status which is temporary).
pollStatus (Text)	The appliance polling status computed from lastPollDate. The appliance is considered as unreachable with a 5 minutes timeout (Timeout can be changed later). The valid values are: POLLING: last poll date is greater than the timeout (5 minutes). NOT_POLLING: the last poll date is less than timeout (5 minutes).
heartbeatGenerated (Date)	Date when the WAF appliance generates a new Heartbeat message. It is not updated directly, but retrieved asynchronously and is synchronized when Qualys Cloud platform/WAF module updates their respective records.

	The Heartbeat message contains system information such as systemOs, systemRam, and systemType.
heartbeatProcessed (Date)	Date when the WAF server processed the Heartbeat message (stored and available). It is not updated directly, but retrieved asynchronously and is synchronized when Qualys Cloud platform/WAF module updates their respective records. The Heartbeat message contains system information such as systemOs, systemRam, and systemType.
systemOs (Text)	The operating system detected on the WAF appliance.
systemRam (Long)	Total RAM detected on the WAF appliance.
systemType (Text)	System deployment type. EC2: for AWS Other: for undefined virtual platforms
systemEc2InstanceId (Text)	Instance identifier of the system on the WAF appliance. For EC2, unique AWS virtual machine identifier. Valid only for EC2 systems
systemEc2InstanceType (Text)	Type of instance. For EC2 type appliance, gives AWS virtual machine scheme. (such as "x-large")

	Valid only for EC2 systems
systemEc2Amild (Text)	For EC2 type appliance, the virtual machine image identifier used to create AWS virtual machine. Valid only for EC2 systems.
systemCpusCount (Long)	Number of CPU (socket) physical or virtual.
systemCpusCores (Long)	Number of core(s) per CPU (socket).
systemCpusSpeed (Float)	CPU frequency in KHz such as 2678.9 for a 2.6 GHz CPU.
systemCpusModel (Text)	The type of CPU model on the WAF appliance. Simple text identifier that describes CPU virtualized (Such as "Intel Bi-Xeon xxxx")
configRulesVersion (Text)	The version of security rules (rules provided by Qualys) that are available on the WAF appliance.
configVersion (Text)	Generated configuration version. The version is not updated directly, but is retrieved asynchronously and is synchronized when WAF module/Qualys Cloud platform update their respective records.
configGenerated (Date)	Date when the configuration is generated on the Qualys Cloud platform.
ip (Text)	The IP address of the WAF appliance. This field may not be configured depending on appliance code, version, and network settings. The appliance may

	not be able to detect its own IP address, if multiple network devices exist.
cluster.id (Long)	Cluster asset identifier within Qualys Cloud platform identifying appliance's group.
cluster.uuid (UUID)	Cluster asset identifier within WAF module identifying the appliance's group.
cluster.name (Text)	Cluster asset name within WAF module identifying the appliance's group.

Sample

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/appliance"

Note: "file.xml" contains the request POST data.
The request POST data is optional. If you leave it empty all
appliances in the user’s scope are returned.
```

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <filters>
    <Criteria field="ip" operator="EQUALS">172.17.0.2</Criteria>
  </filters>
</ServiceRequest>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://<server.host>:<server.port>/qps/
xsd/2.0/waf/appliance.xsd">
  <responseCode>SUCCESS</responseCode>
```

```
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <Appliance>
      <id>15804</id>
      <uuid>9276b0ed-53e4-4c43-b27e-463851990ee3</uuid>
      <name><![CDATA[9276B0ED-53E4-4C43-B27E-463851990EE3]]></name>
      <hostname><![CDATA[30c1def71a9d]]></hostname>
      <lastPollDate>2017-07-25T13:26:30Z</lastPollDate>
      <applianceCreated>2017-07-25T12:51:03Z</applianceCreated>
      <applianceVersion><![CDATA[1.4.0]]></applianceVersion>
      <status>ACTIVE</status>
      <pollStatus>POLLING</pollStatus>
      <heartbeatGenerated>2017-07-25T13:26:00Z</heartbeatGenerated>
      <heartbeatProcessed>2017-07-25T13:26:00Z</heartbeatProcessed>
      <systemOs><![CDATA[Linux 30c1def71a9d 4.4.0-83-generic #106-
Ubuntu SMP Mon Jun 26 17:54:43 UTC 2017 x86_64]]></systemOs>
      <systemRam>12284530688</systemRam>
      <systemType><![CDATA[other]]></systemType>
      <systemCpusCount>1</systemCpusCount>
      <systemCpusCores>2</systemCpusCores>
      <systemCpusSpeed>2600.406</systemCpusSpeed>
      <systemCpusModel><![CDATA[Intel(R) Core(TM) i7-5600U CPU @
2.60GHz]]></systemCpusModel>
     <configRulesVersion><![CDATA[2017.7.14.269]]></configRulesVersion
>
      <configVersion><![CDATA[2017-07-
25T13:18:27.348Z]]></configVersion>
      <configGenerated>2017-07-25T13:18:27Z</configGenerated>
      <ip><![CDATA[172.17.0.2]]></ip>
      <cluster>
        <id>25603</id>
        <uuid>d817eb34-2320-4044-9c0c-8b0b2523a75a</uuid>
        <name><![CDATA[Montlegun]]></name>
      </cluster>
    </Appliance>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/appliance.xsd

Search appliances

/qps/rest/2.0/delete/waf/appliance/<id>

[POST]

Delete a WAF appliance in the user's account.

Permissions required - Managers with full scope. Other users must have WAF module enabled, "API ACCESS" permission, "Manage WAFs" permission, and appliances licensed for WAF and within the user's scope.

Input Parameters

The "id" (Long) element is required. This identifies the appliance you want to delete.

Sample

API request

```
curl -u "USERNAME:PASSWORD" -X "POST" -H "Content-Type: text/xml"
"https://qualysapi.qualys.com/qps/rest/2.0/delete/waf/appliance/15804"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/
xsd/2.0/waf/appliance.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Appliance>
      <id>15804</id>
    </Appliance>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/2.0/waf/appliance.xsd

Events API

Search events

/qps/rest/2.0/search/waf/eventlog

[POST]

Finds security events in the event log using the search filters provided by the 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 the API will return only the current day events. All dates must be entered in UTC date/time format.

Supported filter operators

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	True if the event is flagged.

(Boolean)

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.
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.

Sample

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>
```

Response

```
<?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>
            </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>
```

XSD

<platform API server>/qps/xsd/2.0/waf/eventlog.xsd