Web Application Scanning API

User Guide
Version 3.18

June 10, 2024

Table of Contents

Get Started

Web Application Scanning API

The Web Application Scanning (WAS) API support scanning and reporting on web applications for security risks.

Modules supported

WAS

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

Qualys and the Qualys logo are proprietary trademarks of Qualys, Inc. All other products or names may be trademarks of their respective companies

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/3.0/count/was/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.

Follow the process below to obtain the first two XML pages for an API request. Apply the same logic to get all the next (n+1) pages until all records are returned. This is indicated when <hasMoreRecords>false</hasMoreRecords>.

Sample 1 - Search web apps

Search for web applications that have a name containing the string “Merchant”. The service request in the POST data file “file.xml” defines this search criteria.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
```

```
--data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/webapp" <
file.xml
Note: “file.xml” contains the request POST data.
```

You’ll notice the operator field value is set to 123, which is the value returned in <lastId> of the previous page output. The GREATER operator is a logical “greater than” (it does not mean greater than or equal to).

Request POST data

```
<ServiceRequest>
  <preferences>
    <limitResults>5</limitResults>
  </preferences>
  <filters>
    <Criteria field="name" operator="CONTAINS">Merchant</Criteria>
  </filters>
</ServiceRequest>
```

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

XML response

```
<ServiceResponse ...>
        <responseCode>SUCCESS</responseCode>
        <COUNT>5</COUNT>
        <hasMoreRecords>true</hasMoreRecords>
        <lastId>123</lastId>
        <data>
          <!--here you will find 5 web application records-->
        </data>
</ServiceResponse>
```

Sample 2

To get the next page of results, you need to edit your service request in “file.xml” that will be passed to API request as a POST payload. According to the <lastId> element returned in the first page, you want the next page of results to start with the object ID 124 or greater.

API request

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

You’ll notice the operator field value is set to 123, which is the value returned in <lastId> of the previous page output. The GREATER operator is a logical “greater than” (it does not mean greater than or equal to).

Request POST data

```
<ServiceRequest>
   <filters>
      <Criteria field="name" operator="CONTAINS">Merchant</Criteria>
      <Criteria field="id" operator="GREATER">123</Criteria>
   </filters>
</ServiceRequest>
```

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

```
<ServiceRequest>
  <filters>
    <Criteria> ... </Criteria>
  </filters>
  <preferences>
    <startFromOffset>100</startFromOffset>
    <limitResults>200</limitResults>
  </preferences>
</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.

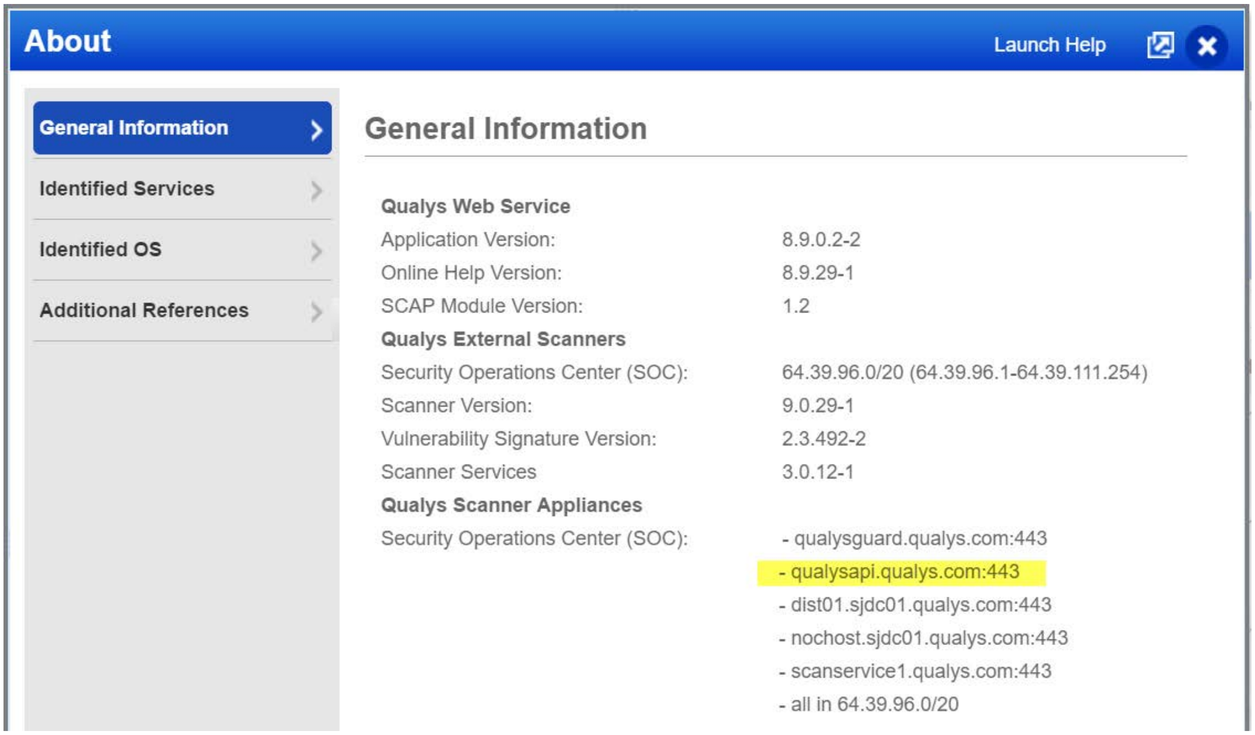
URL to Qualys API server

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

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

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

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).



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

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
```

How to Download Vulnerability Details

/api/2.0/fo/knowledge_base/vuln/?action=list

[GET] [POST]

When you download web application scan results using the WAS API, you'll want to view vulnerability descriptions from the Qualys KnowledgeBase in order to understand the vulnerabilities detected and see our recommended solutions. You can do this programmatically using the KnowledgeBase API (api/2.0/fo/knowledge_base/vuln/?action=list). This API function is part of the Qualys API and it's described in the Qualys API (VM, SCA, PC) User Guide (click here to download the latest version)

Input Parameters

When filter parameters are specified, these parameters are ANDed

Parameter	Description
action=list	(Required) A flag used to request the download of vulnerability data from the KnowledgeBase.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the output.
details={**Basic**|All|None}	(Optional) Show the requested amount of information for each vulnerability in the XML output. A valid value is: Basic (default), All, or None. Basic includes basic elements plus CVSS Base and Temporal scores. All includes all vulnerability details, including the Basic details.

ids={value}	(Optional) Used to filter the XML output to include only vulnerabilities that have QID numbers matching the QID numbers you specify.
id_min={value}	(Optional) Used to filter the XML output to show only vulnerabilities that have a QID number greater than or equal to a QID number you specify.
id_max={value}	(Optional) Used to filter the XML output to show only vulnerabilities that have a QID number less than or equal to a QID number you specify.
is_patchable={0|1}	(Optional) Used to filter the XML output to show only vulnerabilities that are patchable or not patchable. A vulnerability is considered patchable when a patch exists for it. When 1 is specified, only vulnerabilities that are patchable will be included in the output. When 0 is specified, only vulnerabilities that are not patchable will be included in the output. When unspecified, patchable and unpatchable vulnerabilities will be included in the output.
last_modified_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified after a certain date and time. When specified vulnerabilities last modified by a user or by the service will be shown. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_ before={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified before a certain date and time. When specified vulnerabilities last modified by a user or by the service will be shown. The

	date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_by_user_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by a user after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_by_user_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by a user before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_by_service_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by the service after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_by_service_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by the service before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
published_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities published after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
published_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities published before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
discovery_method={value}	(Optional) Used to filter the XML output to show only vulnerabilities assigned a certain discovery method. A valid value is:

	Remote, Authenticated, RemoteOnly, AuthenticatedOnly, or RemoteAndAuthenticated. When “Authenticated” is specified, the service shows vulnerabilities that have at least one associated authentication type. Vulnerabilities that have at least one authentication type can be detected in two ways: 1) remotely without using authentication, and 2) using authentication.
discovery_auth_types={value}	(Optional) Used to filter the XML output to show only vulnerabilities having one or more authentication types. A valid value is: Windows, Oracle, Unix or SNMP. Multiple values are entered as a comma-separated list.
show_pci_reasons={0|1}	(Optional) Used to filter the XML output to show reasons for passing or failing PCI compliance (when the CVSS Scoring feature is turned on in the user’s subscription). Specify 1 to view the reasons in the XML output. When unspecified, the reasons are not included in the XML output.

Sample - All vulnerabilities in KnolwedgeBase, all details

API request

```
curl -u "user:password" -H "X-Requested-With: Curl" -X "POST"
-d "action=list"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" >
output.txt
```

Sample - Patchable vulnerabilities, all details

API request

```
curl -u "user:password" -H "X-Requested-With: Curl" -X "POST"
-d "action=list&ids=1-
200&is_patchable=1&details=All"  "https://qualysapi.qualys.com/api/2.0
/fo/knowledge_base/vuln/" > output.txt
```

Sample - Vulnerabilities modified after certain date

API request

```
curl -u "user:password" -H "X-Requested-With: Curl" -X "POST"
-d "action=list&last_modified_by_service_after=2018-07-20
&discovery_method=RemoteAndAuthenticated"  "https://qualysapi.qualys.c
om/api/2.0/fo/knowledge_base/vuln/" > output.txt
```

DTD

<platform API server>/api/2.0/fo/knowledge_base/vuln/knowledge_base_vuln_list_output.dtd

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>3.5.0.0-SNAPSHOT-1 DEVELOP #92
(2021-01-19T01:51:21Z)</PortalApplication-VERSION>
            <ITAM-VERSION>1.3.1.0-18</ITAM-VERSION>
            <CS-VERSION>1.9.0.0-SNAPSHOT</CS-VERSION>
            <CA-VERSION>3.4.0.0</CA-VERSION>
            <QGS-VERSION>1.2.0.0-6</QGS-VERSION>
            <QUESTIONNAIRE-VERSION>2.26.0.0</QUESTIONNAIRE-VERSION>
            <SAC-VERSION>1.0.0-SNAPSHOT</SAC-VERSION>
            <WAF-VERSION>2.12.6.0</WAF-VERSION>
            <QUESTIONNAIRE__V2-VERSION>1.13.1.0-
SNAPSHOT</QUESTIONNAIRE__V2-VERSION>
            <WAS-VERSION>6.17.0.0-SNAPSHOT-32</WAS-VERSION>
            <FIM-VERSION>2.6.0.0-23</FIM-VERSION>
            <ICS-VERSION>0.9.1.0-12</ICS-VERSION>
            <VM-VERSION>1.0.3</VM-VERSION>
            <CERTVIEW-VERSION>2.8.0.0-20</CERTVIEW-VERSION>
            <CLOUDVIEW-VERSION>1.9.2.0-SNAPSHOT</CLOUDVIEW-VERSION>
            <CM-VERSION>1.31.0.0</CM-VERSION>
            <MDS-VERSION>2.16.1.0-SNAPSHOT-2</MDS-VERSION>
            <PM-VERSION>1.5.0.0-2</PM-VERSION>
            <PS-VERSION>1.3.0.0-16</PS-VERSION>
            <IOC-VERSION>1.2.0-15</IOC-VERSION>
            <THREAT__PROTECT-VERSION>1.5.0-SNAPSHOT</THREAT__PROTECT-
VERSION>
            <AV2-VERSION>0.1.0</AV2-VERSION>
            <UD-VERSION>1.0.0</UD-VERSION>
        </Portal-Version>
        <QWeb-Version>
            <WEB-VERSION>10.7.0.0-1</WEB-VERSION>
            <SCANNER-VERSION>12.1.68-1</SCANNER-VERSION>
            <VULNSIGS-VERSION>2.5.84-2</VULNSIGS-VERSION>
        </QWeb-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": "3.5.0.0-SNAPSHOT-1 DEVELOP #92
(2021-01-19T01:51:21Z)",
          "WAS-VERSION": "6.17.0.0-SNAPSHOT-32",
          "VM-VERSION": "1.0.3",
          "CM-VERSION": "1.20.1",
          "MDS-VERSION": "2.16.1.0-SNAPSHOT-2",
          "CA-VERSION": "2.9.1.0",
          "QUESTIONNAIRE-VERSION": "2.14.0.4",
          "WAF-VERSION": "2.12.6.0"
        },
...
               }
      }
    ],
    "responseCode": "SUCCESS",
    "count": 1
  }
}
```

JSON Support

WAS API supports JSON requests and responses starting with WAS version 4.5. Samples are shown below.

Sample 1 - Create an option profile

API request

```
cat createOP.json | curl -s -X POST -H "Accept: application/json" -H
"Content-Type: application/json" -H "user: username" -H "password:
passwd" -d @-
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/optionprofile/"

POST data:
{
  "ServiceRequest": {
    "data": {
      "OptionProfile": {
        "name": "OP creation - with json request and response",
        "timeoutErrorThreshold": "10",
        "unexpectedErrorThreshold": "20"
      }
    }
  }
}
```

JSON output

```
{
  "ServiceResponse": {
    "data": [
      {
        "OptionProfile": {
          "id": 464134,
          "formSubmission": "BOTH",
          "owner": {
            "lastName": "Smith",
            "username": "username",
            "firstName": "Steve",
            "id": 4354
          },
          "createdBy": {
            "lastName": "Smith",
            "username": "username",
            "firstName": "Steve",
            "id": 4354
          },
          "tags": {
            "count": 0
          },
          "bruteforceOption": "MINIMAL",
          "updatedBy": {
            "lastName": "Smith",
            "username": "username",
            "firstName": "Steve",
            "id": 4354
          },
          "maxCrawlRequests": 300,
          "sensitiveContent": {
            "creditCardNumber": "false",
            "socialSecurityNumber": "false"
          },
          "updatedDate": "2015-12-15T13:39:25Z",
          "comments": {
            "count": 0
          },
          "createdDate": "2015-12-15T13:39:25Z",
          "parameterSet": {
            "name": "Initial Parameters",
            "id": 0
          },
          "isDefault": "false",
          "unexpectedErrorThreshold": 20,
          "performance": "LOW",
          "name": "OP creation - with json request and response",
          "ignoreBinaryFiles": "false",
          "timeoutErrorThreshold": 10
        }
      }
    ],
    "count": 1,
    "responseCode": "SUCCESS"
  }
}
```

Sample 2 - Launch a scan

API request

```
cat createOP.json | curl -s -X POST -H "Accept: application/json" -H
"Content-Type: application/json" -H "user: username" -H "password:
passwd" -d @-
"https://qualysapi.qualys.com/qps/rest/3.0/launch/was/wasscan/"

POST data:
{
  "ServiceRequest": {
    "data": {
      "WasScan": {
        "name": "WebApp Default Auth",
        "type": "VULNERABILITY",
        "target": {
          "webApp": { "id": "2640672" },
          "webAppAuthRecord": { "isDefault": "true" }
        },
        "cancelAfterNHours": "1",
        "profile": { "id": "450936" }
      }
    }
  }
}
```

JSON output

```
{
  "ServiceResponse" : {
    "responseCode" : "SUCCESS",
    "data" : [ {
      "WasScan" : {
        "id" : 1498381
      }
    } ],
    "count" : 1
  }
```

Sample 3 - Get a WAS scan

API request

```
cat createOP.json | curl -s -X POST -H "Accept: application/json" -H
"Content-Type: application/json" -H "user: username" -H "password:
passwd" -d @-
"https://qualysapi.qualys.com/qps/rest/3.0/launch/was/wasscan/"

POST data:
{
  "ServiceRequest": {
    "data": {
      "WasScan": {
        "name": "WebApp Default Auth",
        "type": "VULNERABILITY",
        "target": {
          "webApp": { "id": "2640672" },
          "webAppAuthRecord": { "isDefault": "true" }
        },
        "cancelAfterNHours": "1",
        "profile": { "id": "450936" }
      }
    }
  }
}
```

JSON output

```
{
  "ServiceResponse" : {
    "responseCode" : "SUCCESS",
    "data" : [ {
      "WasScan" : {
        "id" : 1498381
      }
    } ],
    "count" : 1
  }
```

Sample 4 - Search WAS Findings with Multiple Criteria

API request

```
cat createOP.json | curl -s -X POST -H "Accept: application/json" -H
"Content-Type: application/json" -H "user: username" -H "password:
```

```
passwd" -d @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/finding"

POST data:
{
  "ServiceRequest": {
    "preferences": {
      "verbose": "true",
      "limitResults": "2"
    },
    "filters": {
      "Criteria": [
        {
          "field": "id",
          "operator": "EQUALS",
          "value": "3615376"
        },
        {
          "field": "qid",
          "operator": "NOT EQUALS",
          "value": "0"
        }
      ]
    }
  }
}
```

JSON output

```
{
  "ServiceResponse": {
    "data": [
      {
        "Finding": {
          "url": "http://10.11.68.95/bricks/config/",
          "lastDetectedDate": "2021-06-21T02:10:15Z",
          "cwe": {
            "count": 1,
            "list": [
              23
            ]
          },
          "id": 3615376,
          "lastTestedDate": "2021-06-21T02:10:15Z",
          "firstDetectedDate": "2021-06-21T02:10:15Z",
          "findingType": "QUALYS",
          "updatedDate": "2021-06-21T02:26:31Z",
          "history": {
            "set": [
              {
                "WebAppFindingHistory": {
                  "scanData": {
                    "reference": "was/1624029515335.1191085.70",
                    "launchedDate": "2021-06-21T02:10:15Z",
                    "id": 4255627
                  }
                }
              }
            ]
          },
          "potential": "false",
          "status": "NEW",
          "severity": "1",
          "webApp": {
            "id": 8777442,
            "tags": {
              "count": 0
            },
            "url": "http://10.11.68.95/digestApp",
            "name": "Latest Target612"
          },
          "uniqueId": "0bfd3ee4-db6f-4d82-b970-1650a4186637",
          "name": "Path-relative stylesheet import (PRSSI)
vulnerability",
          "qid": 150246,
          "cvssV3": {
            "temporal": 2.9,
            "attackVector": "Network",
            "base": 3.1
          },
          "resultList": {
            "count": 1,
            "list": [
              {
                "Result": {
                  "ajax": "false",
                  "payloads": {
                    "count": 1,
                    "list": [
                      {
                        "PayloadInstance": {
                          "request": {
                            "headers":
"UmVmZXJlcjogaHR0cDovLzEwLjExLjY4Ljk1L2RpZ2VzdEFwcA0KQ29va2llOiBQSFBTR
VNTSUQ9bzAxNm5hMWpnZXZhNmF2OTltdWwxcjRrdDM7DQpIb3N0OiAxMC4xMS42OC45NQ0
KVXNlci1BZ2VudDogTW96aWxsYS81LjAgKE1hY2ludG9zaDsgSW50ZWwgTWFjIE9TIFggM
TBfMTRfNSkgQXBwbGVXZWJLaXQvNjA1LjEuMTUgKEtIVE1MLCBsaWtlIEdlY2tvKSBWZXJ
zaW9uLzEyLjEuMSBTYWZhcmkvNjA1LjEuMTUNCkFjY2VwdDogKi8qDQo=",
                            "method": "GET",
                            "link":
"http://10.11.68.95/bricks/config/"
                          },
                          "response": "\nRelative Path CSS Links
found:\n<link rel=\"stylesheet\"
href=\"../stylesheets/foundation.css\">\n<link rel=\"stylesheet\"
href=\"../stylesheets/foundation.min.css\">\n<link rel=\"stylesheet\"
href=\"../stylesheets/app.css\">",
                          "payload": "N/A"
                        }
                      }
                    ]
                  },
                  "authentication": "false"
                }
              }
            ]
          },
          "isIgnored": "false",
          "timesDetected": 1,
          "type": "VULNERABILITY"
        }
      }
    ],
    "responseCode": "SUCCESS",
    "hasMoreRecords": "false",
    "count": 1
  }
}
```

YAML Support

WAS API supports YAML requests and responses starting with WAS version 8.17. Sample is shown below.

Sample 1 - Create a web application

API request

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

POST data:
<ServiceRequest>
<data>
<WebApp>
<name><![CDATA[TEST YAML API]]></name>
<url>http://10.12.14.16</url>
<swaggerFile>
<name>ajax.yml</name>
<content>LS0tCnN3YWdnZXI6ICcyLjAnCmluZm86CiAgZGVzY3JpcHRpb246IFRoaXMga
XMgYSBzYW1wbGUgUkVTVCBhcGkgc2VydmVyLgogIHZlcnNpb246IDEuMC4wCiAgdGl0bGU
6IFN3YWdnZXIgV2l0aCBWdWxuZXJhYmlsaXR5CiAgdGVybXNPZlNlcnZpY2U6IGh0dHA6L
y9zd2FnZ2VyLmlvL3Rlcm1zLwogIGNvbnRhY3Q6CiAgICBlbWFpbDogYWJjQGd4bWFpbC5
jb20KICBsaWNlbnNlOgogICAgbmFtZTogQXBhY2hlIDIuMAogICAgdXJsOiBodHRwOi8vd
3d3LmFwYWNoZS5vcmcvbGljZW5zZXMvTElDRU5TRS0yLjAuaHRtbApob3N0OiAxMC4xMS4
2OS4yMQpiYXNlUGF0aDogIi9KU09OIgp0YWdzOgotIG5hbWU6IHNlYXJjaAogIGRlc2Nya
XB0aW9uOiBTZWFyY2gKICBleHRlcm5hbERvY3M6CiAgICBkZXNjcmlwdGlvbjogRmluZCB
vdXQgbW9yZQogICAgdXJsOiBodHRwOi8vc3dhZ2dlci5pbwotIG5hbWU6IHNlYXJjaDIKI
CBkZXNjcmlwdGlvbjogQWNjZXNzIHRvIFBldHN0b3JlIG9yZGVycwotIG5hbWU6IHVzZXI
KICBkZXNjcmlwdGlvbjogT3BlcmF0aW9ucyBhYm91dCB1c2VyCiAgZXh0ZXJuYWxEb2NzO
gogICAgZGVzY3JpcHRpb246IEZpbmQgb3V0IG1vcmUgYWJvdXQgb3VyIHN0b3JlCiAgICB
1cmw6IGh0dHA6Ly9zd2FnZ2VyLmlvCnNjaGVtZXM6Ci0gaHR0cApwYXRoczoKICAiLzIvY
WpheF9zZWFyY2hfMS5waHAiOgogICAgcG9zdDoKICAgICAgdGFnczoKICAgICAgLSBzZWF
yY2gKICAgICAgc3VtbWFyeTogU2VhcmNoIGZvciBoaXN0b3J5CiAgICAgIGRlc2NyaXB0a
W9uOiAnJwogICAgICBvcGVyYXRpb25JZDogc2VhcmNoMQogICAgICBjb25zdW1lczoKICA
```

```
gICAgLSBhcHBsaWNhdGlvbi9qc29uCiAgICAgIHByb2R1Y2VzOgogICAgICAtIHRleHQva
HRtbAogICAgICBwYXJhbWV0ZXJzOgogICAgICAtIGluOiBib2R5CiAgICAgICAgbmFtZTo
gYm9keQogICAgICAgIGRlc2NyaXB0aW9uOiBTZWFyY2ggcGFyYW1ldGVycwogICAgICAgI
HJlcXVpcmVkOiB0cnVlCiAgICAgICAgc2NoZW1hOgogICAgICAgICAgIiRyZWYiOiAiIy9
kZWZpbml0aW9ucy9TZWFyY2gxIgogICAgICByZXNwb25zZXM6CiAgICAgICAgJzIwMCc6C
iAgICAgICAgICBkZXNjcmlwdGlvbjogU3VjY2VzcwogICAgICAgICc0MTUnOgogICAgICA
gICAgZGVzY3JpcHRpb246IFVuc3VwcG9ydGVkIG1lZGlhCiAgICAgICAgJzQyOSc6CiAgI
CAgICAgICBkZXNjcmlwdGlvbjogVG9vIG1hbnkgcmVxdWVzdHMKICAgICAgICBkZWZhdWx
0OgogICAgICAgICAgZGVzY3JpcHRpb246IERlZmF1bHQgZXJyb3IgcmVzcG9uc2UKICAiL
zIvYWpheF9zZWFyY2hfMi5waHAiOgogICAgcG9zdDoKICAgICAgdGFnczoKICAgICAgLSB
zZWFyY2gyCiAgICAgIHN1bW1hcnk6IFNlYXJjaCBmb3IgaGlzdG9yeTIKICAgICAgb3Blc
mF0aW9uSWQ6IHNlYXJjaDIKICAgICAgY29uc3VtZXM6CiAgICAgIC0gYXBwbGljYXRpb24
vanNvbgogICAgICBwcm9kdWNlczoKICAgICAgLSB0ZXh0L2h0bWwKICAgICAgcGFyYW1ld
GVyczoKICAgICAgLSBpbjogYm9keQogICAgICAgIG5hbWU6IGJvZHkKICAgICAgICBkZXN
jcmlwdGlvbjogU2VhcmNoIHBhcmFtZXRlcnMKICAgICAgICByZXF1aXJlZDogdHJ1ZQogI
CAgICAgIHNjaGVtYToKICAgICAgICAgICIkcmVmIjogIiMvZGVmaW5pdGlvbnMvU2VhcmN
oMiIKICAgICAgcmVzcG9uc2VzOgogICAgICAgICcyMDAnOgogICAgICAgICAgZGVzY3Jpc
HRpb246IFN1Y2Nlc3MKICAgICAgICAnNDE1JzoKICAgICAgICAgIGRlc2NyaXB0aW9uOiB
VbnN1cHBvcnRlZCBtZWRpYQogICAgICAgICc0MjknOgogICAgICAgICAgZGVzY3JpcHRpb
246IFRvbyBtYW55IHJlcXVlc3RzCiAgICAgICAgZGVmYXVsdDoKICAgICAgICAgIGRlc2N
yaXB0aW9uOiBEZWZhdWx0IGVycm9yIHJlc3BvbnNlCmRlZmluaXRpb25zOgogIFNlYXJja
DE6CiAgICB0eXBlOiBvYmplY3QKICAgIHByb3BlcnRpZXM6CiAgICAgIGxpbWl0OgogICA
gICAgIHR5cGU6IGludGVnZXIKICAgICAgICBmb3JtYXQ6IGludDY0CiAgICAgICAgbWF4a
W11bTogMjAwMAogICAgICAgIG1pbmltdW06IDEwCiAgICAgIG9yZGVyOgogICAgICAgIHR
5cGU6IHN0cmluZwogICAgICAgIGVudW06CiAgICAgICAgLSBhc2MKICAgICAgICAtIGRlc
2MKICAgICAgdGVybToKICAgICAgICB0eXBlOiBzdHJpbmcKICAgICAgICBtaW5MZW5ndGg
6IDEKICAgICAgICBtYXhMZW5ndGg6IDIwCiAgICAgICAgcGF0dGVybjogIlthLXpBLVowL
TlfXSIKICBTZWFyY2gyOgogICAgdHlwZTogb2JqZWN0CiAgICBwcm9wZXJ0aWVzOgogICA
gICBsaW1pdDI6CiAgICAgICAgdHlwZTogaW50ZWdlcgogICAgICAgIGZvcm1hdDogaW50N
jQKICAgICAgICBtYXhpbXVtOiAyMDAwCiAgICAgICAgbWluaW11bTogMTAKICAgICAgb3J
kZXIyOgogICAgICAgIHR5cGU6IHN0cmluZwogICAgICAgIGVudW06CiAgICAgICAgLSBhc
2MKICAgICAgICAtIGRlc2MKICAgICAgdGVybTI6CiAgICAgICAgdHlwZTogc3RyaW5nCiA
gICAgICAgbWluTGVuZ3RoOiAxCiAgICAgICAgbWF4TGVuZ3RoOiAyMAogICAgICAgIHBhd
HRlcm46ICJbYS16QS1aMC05X10iCiAgICAgIG9mZnNldDI6CiAgICAgICAgdHlwZTogaW5
0ZWdlcgogICAgICAgIGZvcm1hdDogaW50NjQKICAgICAgICBtYXhpbXVtOiAyOTM4MjQ3M
zIKICAgICAgICBtaW5pbXVtOiAxCmV4dGVybmFsRG9jczoKICBkZXNjcmlwdGlvbjogRml
uZCBvdXQgbW9yZSBhYm91dCBTd2FnZ2VyCiAgdXJsOiBodHRwOi8vc3dhZ2dlci5pbwo=<
/content>
</swaggerFile>
</WebApp>
</data>
</ServiceRequest>
```

YAML output

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

<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/
qps/xsd/3.0/was/webapp.xsd">
<responseCode>SUCCESS</responseCode>
<count>1</count>
<data>
<WebApp>
<id>14763246</id>
<name>
<![CDATA[TEST YAML API]]>
</name>
<url>
<![CDATA[http://10.12.14.16]]>
</url>
<owner>
<id>8296038</id>
<username>quays_abc</username>
<firstName>
<![CDATA[John]]>
</firstName>
<lastName>
<![CDATA[Doe]]>
</lastName>
</owner>
<scope>ALL</scope>
<attributes>
<count>0</count>
</attributes>
<defaultScanner>
<type>EXTERNAL</type>
</defaultScanner>
<scannerLocked>false</scannerLocked>
<progressiveScanning>DISABLED</progressiveScanning>
<urlExcludelist>
<count>0</count>
</urlExcludelist>
<urlAllowlist>
<count>0</count>
</urlAllowlist>
<postDataExcludelist>
<count>0</count>
</postDataExcludelist>
<logoutRegexList>
```

```
<count>0</count>
</logoutRegexList>
<authRecords>
<count>0</count>
</authRecords>
<dnsOverrides>
<count>0</count>
</dnsOverrides>
<useRobots>IGNORE</useRobots>
<useSitemap>false</useSitemap>
<malwareMonitoring>false</malwareMonitoring>
<malwareNotification>false</malwareNotification>
<tags>
<count>0</count>
</tags>
<comments>
<count>0</count>
</comments>
<isScheduled>false</isScheduled>
<createdBy>
<id>8296038</id>
<username>quays_abc</username>
<firstName>
<![CDATA[John]]>
</firstName>
<lastName>
<![CDATA[Doe]]>
</lastName>
</createdBy>
<createdDate>2022-06-02T09:27:50Z</createdDate>
<updatedBy>
<id>8296038</id>
<username>quays_abc</username>
<firstName>
<![CDATA[John]]>
</firstName>
<lastName>
<![CDATA[Doe]]>
</lastName>
</updatedBy>
<updatedDate>2022-06-02T09:27:50Z</updatedDate>
<config/>
<crawlingScripts>
<count>0</count>
</crawlingScripts>
```

```
<swaggerFile>
<id>10291</id>
<name>ajax.yml</name>
<content>LS0tCnN3YWdnZXI6ICcyLjAnCmluZm86CiAgZGVzY3JpcHRpb246IFRoaXMga
XMgYSBzYW1wbGUgUkVTVCBhcGkgc2VydmVyLgogIHZlcnNpb246IDEuMC4wCiAgdGl0bGU
6IFN3YWdnZXIgV2l0aCBWdWxuZXJhYmlsaXR5CiAgdGVybXNPZlNlcnZpY2U6IGh0dHA6L
y9zd2FnZ2VyLmlvL3Rlcm1zLwogIGNvbnRhY3Q6CiAgICBlbWFpbDogYWJjQGd4bWFpbC5
jb20KICBsaWNlbnNlOgogICAgbmFtZTogQXBhY2hlIDIuMAogICAgdXJsOiBodHRwOi8vd
3d3LmFwYWNoZS5vcmcvbGljZW5zZXMvTElDRU5TRS0yLjAuaHRtbApob3N0OiAxMC4xMS4
2OS4yMQpiYXNlUGF0aDogIi9KU09OIgp0YWdzOgotIG5hbWU6IHNlYXJjaAogIGRlc2Nya
XB0aW9uOiBTZWFyY2gKICBleHRlcm5hbERvY3M6CiAgICBkZXNjcmlwdGlvbjogRmluZCB
vdXQgbW9yZQogICAgdXJsOiBodHRwOi8vc3dhZ2dlci5pbwotIG5hbWU6IHNlYXJjaDIKI
CBkZXNjcmlwdGlvbjogQWNjZXNzIHRvIFBldHN0b3JlIG9yZGVycwotIG5hbWU6IHVzZXI
KICBkZXNjcmlwdGlvbjogT3BlcmF0aW9ucyBhYm91dCB1c2VyCiAgZXh0ZXJuYWxEb2NzO
gogICAgZGVzY3JpcHRpb246IEZpbmQgb3V0IG1vcmUgYWJvdXQgb3VyIHN0b3JlCiAgICB
1cmw6IGh0dHA6Ly9zd2FnZ2VyLmlvCnNjaGVtZXM6Ci0gaHR0cApwYXRoczoKICAiLzIvY
WpheF9zZWFyY2hfMS5waHAiOgogICAgcG9zdDoKICAgICAgdGFnczoKICAgICAgLSBzZWF
yY2gKICAgICAgc3VtbWFyeTogU2VhcmNoIGZvciBoaXN0b3J5CiAgICAgIGRlc2NyaXB0a
W9uOiAnJwogICAgICBvcGVyYXRpb25JZDogc2VhcmNoMQogICAgICBjb25zdW1lczoKICA
gICAgLSBhcHBsaWNhdGlvbi9qc29uCiAgICAgIHByb2R1Y2VzOgogICAgICAtIHRleHQva
HRtbAogICAgICBwYXJhbWV0ZXJzOgogICAgICAtIGluOiBib2R5CiAgICAgICAgbmFtZTo
gYm9keQogICAgICAgIGRlc2NyaXB0aW9uOiBTZWFyY2ggcGFyYW1ldGVycwogICAgICAgI
HJlcXVpcmVkOiB0cnVlCiAgICAgICAgc2NoZW1hOgogICAgICAgICAgIiRyZWYiOiAiIy9
kZWZpbml0aW9ucy9TZWFyY2gxIgogICAgICByZXNwb25zZXM6CiAgICAgICAgJzIwMCc6C
iAgICAgICAgICBkZXNjcmlwdGlvbjogU3VjY2VzcwogICAgICAgICc0MTUnOgogICAgICA
gICAgZGVzY3JpcHRpb246IFVuc3VwcG9ydGVkIG1lZGlhCiAgICAgICAgJzQyOSc6CiAgI
CAgICAgICBkZXNjcmlwdGlvbjogVG9vIG1hbnkgcmVxdWVzdHMKICAgICAgICBkZWZhdWx
0OgogICAgICAgICAgZGVzY3JpcHRpb246IERlZmF1bHQgZXJyb3IgcmVzcG9uc2UKICAiL
zIvYWpheF9zZWFyY2hfMi5waHAiOgogICAgcG9zdDoKICAgICAgdGFnczoKICAgICAgLSB
zZWFyY2gyCiAgICAgIHN1bW1hcnk6IFNlYXJjaCBmb3IgaGlzdG9yeTIKICAgICAgb3Blc
mF0aW9uSWQ6IHNlYXJjaDIKICAgICAgY29uc3VtZXM6CiAgICAgIC0gYXBwbGljYXRpb24
vanNvbgogICAgICBwcm9kdWNlczoKICAgICAgLSB0ZXh0L2h0bWwKICAgICAgcGFyYW1ld
GVyczoKICAgICAgLSBpbjogYm9keQogICAgICAgIG5hbWU6IGJvZHkKICAgICAgICBkZXN
jcmlwdGlvbjogU2VhcmNoIHBhcmFtZXRlcnMKICAgICAgICByZXF1aXJlZDogdHJ1ZQogI
CAgICAgIHNjaGVtYToKICAgICAgICAgICIkcmVmIjogIiMvZGVmaW5pdGlvbnMvU2VhcmN
oMiIKICAgICAgcmVzcG9uc2VzOgogICAgICAgICcyMDAnOgogICAgICAgICAgZGVzY3Jpc
HRpb246IFN1Y2Nlc3MKICAgICAgICAnNDE1JzoKICAgICAgICAgIGRlc2NyaXB0aW9uOiB
VbnN1cHBvcnRlZCBtZWRpYQogICAgICAgICc0MjknOgogICAgICAgICAgZGVzY3JpcHRpb
246IFRvbyBtYW55IHJlcXVlc3RzCiAgICAgICAgZGVmYXVsdDoKICAgICAgICAgIGRlc2N
yaXB0aW9uOiBEZWZhdWx0IGVycm9yIHJlc3BvbnNlCmRlZmluaXRpb25zOgogIFNlYXJja
DE6CiAgICB0eXBlOiBvYmplY3QKICAgIHByb3BlcnRpZXM6CiAgICAgIGxpbWl0OgogICA
gICAgIHR5cGU6IGludGVnZXIKICAgICAgICBmb3JtYXQ6IGludDY0CiAgICAgICAgbWF4a
W11bTogMjAwMAogICAgICAgIG1pbmltdW06IDEwCiAgICAgIG9yZGVyOgogICAgICAgIHR
5cGU6IHN0cmluZwogICAgICAgIGVudW06CiAgICAgICAgLSBhc2MKICAgICAgICAtIGRlc
2MKICAgICAgdGVybToKICAgICAgICB0eXBlOiBzdHJpbmcKICAgICAgICBtaW5MZW5ndGg
```

```
6IDEKICAgICAgICBtYXhMZW5ndGg6IDIwCiAgICAgICAgcGF0dGVybjogIlthLXpBLVowL
TlfXSIKICBTZWFyY2gyOgogICAgdHlwZTogb2JqZWN0CiAgICBwcm9wZXJ0aWVzOgogICA
gICBsaW1pdDI6CiAgICAgICAgdHlwZTogaW50ZWdlcgogICAgICAgIGZvcm1hdDogaW50N
jQKICAgICAgICBtYXhpbXVtOiAyMDAwCiAgICAgICAgbWluaW11bTogMTAKICAgICAgb3J
kZXIyOgogICAgICAgIHR5cGU6IHN0cmluZwogICAgICAgIGVudW06CiAgICAgICAgLSBhc
2MKICAgICAgICAtIGRlc2MKICAgICAgdGVybTI6CiAgICAgICAgdHlwZTogc3RyaW5nCiA
gICAgICAgbWluTGVuZ3RoOiAxCiAgICAgICAgbWF4TGVuZ3RoOiAyMAogICAgICAgIHBhd
HRlcm46ICJbYS16QS1aMC05X10iCiAgICAgIG9mZnNldDI6CiAgICAgICAgdHlwZTogaW5
0ZWdlcgogICAgICAgIGZvcm1hdDogaW50NjQKICAgICAgICBtYXhpbXVtOiAyOTM4MjQ3M
zIKICAgICAgICBtaW5pbXVtOiAxCmV4dGVybmFsRG9jczoKICBkZXNjcmlwdGlvbjogRml
uZCBvdXQgbW9yZSBhYm91dCBTd2FnZ2VyCiAgdXJsOiBodHRwOi8vc3dhZ2dlci5pbwo=<
/content>
<fileSize>2774</fileSize>
</swaggerFile>
</WebApp>
</data>
</ServiceResponse>
```

Web Applications

Web Application Count

/qps/rest/3.0/count/was/webapp

[GET] [POST]

Returns the total number of web applications in the user's account. Input elements are optional and are used to filter the number of web applications included in the count.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The count includes web applications in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements.

Click here for available operators

Parameter	Description
id	(integer) Web application ID.
name	(text) Web application name.
url	(text) The URL of web application.
tags.name	(text) Tag name assigned to web application.
tags.id	(integer) Tag ID assigned to web application.

createdDate	(date) The date when the web application was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the web application was last updated in WAS, in UTC date/time format.
isScheduled	(boolean) A flag indicating whether a scan is scheduled for web application.
isScanned	(boolean) A flag indicating whether the web application has been scanned.
lastScan.status	(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED
lastScan.date	(date) Date when web application was last scanned, in UTC date/time format.

Sample - Get count of web apps, all in user's account

API request

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/webapp"
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>227</count>
</ServiceResponse>
```

Sample - Get count of web apps in ID range

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="IN">323126,323816</Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>0</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/webapp.xsd

Search Web Application

/qps/rest/3.0/search/was/webapp

[POST]

Returns a list of web applications which are in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes web applications in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements.

The special field=attributes attribute for the Criteria element is used to search custom attributes (see sample below).

Click here for available operators

Parameter	Description
id	(integer) Web application ID.
name	(text) Web application name.
url	(text) The URL of web application.
tags	(element) Tags assigned to web application. Click here for description of this <WebApp> element
tags.name	(text) Tag name assigned to web application.
tags.id	(integer) Tag ID assigned to web application.

createdDate	(date) The date when the web application was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the web application was last updated in WAS, in UTC date/time format.
isScheduled	(boolean) A flag indicating whether a scan is scheduled for web application.
isScanned	(boolean) A flag indicating whether the web application has been scanned.
lastScan.status	(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED
lastScan.date	(date) Date when web application was last scanned, in UTC date/time format.
verbose	(boolean) A flag to indicate whether the list of tags associated with the web application should be listed or not. Example: <preferences> <verbose>true</verbose> </preferences>

Sample - List all web apps in user's account

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type:
text/xml"  "https://qualysapi.qualys.com/qps/rest/3.0/search/was/webap
p" -X "POST"
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <hasMoreRecords>false</hasMoreRecords>
  <lastId>323103</lastId>
  <data>
    <WebApp>
      <id>323102</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[https://example.com]]></url>
      <owner>
        <id>123068</id>
      </owner>
      <tags>
        <count>3</count>
      </tags>
      <createdDate>2017-11-22T13:48:03Z</createdDate>
      <updatedDate>2018-09-19T13:41:07Z</updatedDate>
    </WebApp>
    <WebApp>
      <id>323103</id>
      <name><![CDATA[Demo Web App]]></name>
      <url><![CDATA[http://10.10.26.200:80/phpBB/1.4.4_basic]]></url>
      <owner>
        <id>123071</id>
      </owner>
      <tags>
        <count>0</count>
      </tags>
      <createdDate>2018-06-22T13:45:46Z</createdDate>
      <updatedDate>2018-09-16T14:33:38Z</updatedDate>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - List certain web apps

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
```

```
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/webapp" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
 <filters>
   <Criteria field="name" operator="CONTAINS">Merchant</Criteria>
   <Criteria field="id" operator="GREATER">323000</Criteria>
 </filters>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WebApp>
      <id>323476</id>
      <name><![CDATA[Merchant site 1]]></name>
      <url><![CDATA[http://10.10.25.116:80/merchant/2.2/themerchant]]>
</url>
      <owner>
        <id>123056</id>
      </owner>
      <tags>
        <count>0</count>
      </tags>
      <createdDate>2018-02-21T15:24:49Z</createdDate>
      <updatedDate>2018-07-03T16:53:37Z</updatedDate>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Search Web Application and view associated tags

API request

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

Request POST data

```
<ServiceRequest>
    <preferences>
        <verbose>true</verbose>
    </preferences>
    <filters>
        <Criteria field="name" operator="CONTAINS">My Web
Application</Criteria>
    </filters>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <WebApp>
            <id>6620298</id>
            <name>
                <![CDATA[My Web Application]]>
            </name>
            <url>
                <![CDATA[http://www.example.com]]>
            </url>
            <owner>
                <id>1056860</id>
            </owner>
            <tags>
                <count>1</count>
                <list>
                   <Tag>
                       <id>9029017</id>
                       <name>
```

```
                           <![CDATA[TagWebapp1]]>
                       </name>
                   </Tag>
                </list>
            </tags>
            <createdDate>2017-12-15T16:13:06Z</createdDate>
            <updatedDate>2018-11-19T04:38:08Z</updatedDate>
        </WebApp>
    </data>
</ServiceResponse>
```

Sample - Search custom attributes

Search custom attributes using the field attribute for the Criteria element.

API request

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

Find web applications that have a custom attribute name “Function” and this attribute has a value that contains “web” (case insensitive search).

Request POST data

```
<ServiceRequest>
       <filters>
         <Criteria field="attributes"
name="Function"  operator="CONTAINS">web</Criteria>
       </filters>
</ServiceRequest>
```

Find web applications that have a custom attribute name “Function” and this attribute has a value that is equal to “web”.

Request POST data (EQUALS)

```
<ServiceRequest>
       <filters>
```

```
         <Criteria field="attributes" name="Function"
operator="EQUALS">web</Criteria>
       </filters>
</ServiceRequest>
```

Find web applications that have a custom attribute name “Function” and this attribute has a value not equal to “web”.

Request POST data (NOT EQUALS)

```
<ServiceRequest>
       <filters>
         <Criteria field="attributes" name="Function" operator="NOT
EQUALS">web</Criteria>
       </filters>
</ServiceRequest>
```

XSD

<platform API server>/qps/xsd/3.0/was/webapp.xsd

Get Web Application Details

/qps/rest/3.0/get/was/webapp/<id>

[GET]

Returns details for a web application which is in the user's scope. Want to find a web application ID to use as input? See Search Web applications.

The web application screenshot, when available, is included in the output in the "screenshot" element as a base64 encoded binary string. This string needs to be converted before a user can decode and view the image file (.jpg).

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes web applications in the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies a web application.

Click here for available operators

Samples

View details for the web application

Get details - DNS override settings

Get details - logout regular expression list

View default authentication record details

Get details - Selenium crawl script

Get details of a progressive scan

Sample - View details for the web application

Let us view details for the web application with the ID 2130421.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webapp/2130421"
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>2130421</id>
      <name><![CDATA[CUSTOM PARAM TEST]]></name>
      <url><![CDATA
[http://funkytown.abcd01.abcd.com/Forms/FormFields/temp/]]></url>
      <os>Linux 2.4-2.6 / Embedded Device / F5 Networks Big-IP / Linux
2.6</os>
      <owner>
        <id>4354</id>
        <username>user_alex</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <scope>ALL</scope>
      <attributes>
        <count>0</count>
      </attributes>
      <defaultProfile>
        <id>139359</id>
        <name><![CDATA[10 Links edit]]></name>
      </defaultProfile>
      <defaultScanner>
        <type>EXTERNAL</type>
      </defaultScanner>
      <scannerLocked>false</scannerLocked>
      <urlExcludelist>
        <count>0</count>
      </urlExcludelist>
      <urlAllowlist>
        <count>0</count>
      </urlAllowlist>
      <postDataExcludelist>
        <count>0</count>
      </postDataExcludelist>
      <authRecords>
        <count>1</count>
        <list>
          <WebAppAuthRecord>
            <id>127357</id>
            <name><![CDATA[AR - funkytown]]></name>
          </WebAppAuthRecord>
        </list>
      </authRecords>
      <useRobots>IGNORE</useRobots>
      <useSitemap>false</useSitemap>
      <malwareMonitoring>true</malwareMonitoring>
      <malwareNotification>true</malwareNotification>
      <malwareScheduling>
        <startDate>2017-03-03T09:50:00Z</startDate>
        <timeZone>
          <code>Asia/Kolkata</code>
          <offset>+05:30</offset>
        </timeZone>
        <occurrenceType>MONTHLY</occurrenceType>
        <occurrence>
          <monthlyOccurrence>
            <monthlyType>
              <occurDayOrderInMonth>
                <dayOrder>FIRST</dayOrder>
                <dayOfMonth>THURSDAY</dayOfMonth>
                <everyNMonths>1</everyNMonths>
              </occurDayOrderInMonth>
            </monthlyType>
            <occurrenceCount>4</occurrenceCount>
          </monthlyOccurrence>
        </occurrence>
      </malwareScheduling>
      <tags>
        <count>4</count>
        <list>
          <Tag>
            <id>1730872</id>
            <name><![CDATA[new tag]]></name>
          </Tag>
          <Tag>
            <id>1418973</id>
            <name><![CDATA[Cert Tag]]></name>
          </Tag>
          <Tag>
            <id>1693034</id>
            <name><![CDATA[My Tag name]]></name>
          </Tag>
          <Tag>
            <id>1693032</id>
            <name><![CDATA[Groovy tag -1]]></name>
          </Tag>
        </list>
      </tags>
      <comments>
        <count>0</count>
      </comments>
      <isScheduled>false</isScheduled>
      <lastScan>
        <id>827468</id>
        <name><![CDATA[Web Application Vulnerability Scan - CUSTOM
PARAM TEST]]></name>
      </lastScan>
      <createdBy>
        <id>4354</id>
        <username>user_alex</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <createdDate>2017-07-24T09:08:49Z</createdDate>
      <updatedBy>
        <id>4354</id>
        <username>user_alex</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
      <updatedDate>2017-09-24T23:34:17Z</updatedDate>
<screenshot><![CDATA[_9j_4AAQSkZJRgABAQEAegBrAAD_2wBDAAYEBQYFBAYGBQYHB
wYIChAKCgkJChQODwwQFxQYGBcUFhYaHSUfGhsjHBYWICwgIyYnKSopGR8tMC0oMCUoKSj
_2wBDAQcHBwoIChMKChMoGhYaKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCg
… (shortened for brevity)
Convert this value in order to decode and view the image file (.jpg).
</screenshot>
```

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

Sample - Get details - DNS override settings

Let us get details of the web application with ID 2508873 that includes DNS override records. The dnsOverrides element lists the records.

API request

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webapp/2508873"
```

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/3.
0/was/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>2508873</id>
            <name>
                <![CDATA[My Web App]]>
            </name>
            <url>          <![CDATA[http://funkytown.vuln.qa.com:80/ca
ssium/xss/]]>
            </url>
            <owner>
                <id>4354</id>
                <username>user_adam</username>
                <firstName>
                    <![CDATA[Adam]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </owner>
            <scope>ALL</scope>
            <attributes>
                <count>0</count>
            </attributes>
            <defaultScanner>
                <type>INTERNAL</type>
                <friendlyName>
                    <![CDATA[db4_abcd_ab2]]>
                </friendlyName>
            </defaultScanner>
            <scannerLocked>false</scannerLocked>
            <progressiveScanning>ENABLED</progressiveScanning>
            <urlExcludelist>
                <count>0</count>
            </urlExcludelist>
            <urlAllowlist>
                <count>0</count>
            </urlAllowlist>
            <postDataExcludelist>
                <count>0</count>
            </postDataExcludelist>
            <authRecords>
                <count>0</count>
            </authRecords>
            <dnsOverrides>
                <count>2</count>
                <list>
                    <DnsOverride>
                        <id>1620</id>
                        <name>
                            <![CDATA[DNS Override Settings 1]]>
                        </name>
                    </DnsOverride>
                    <DnsOverride>
                        <id>1020</id>
                        <name>
                            <![CDATA[DNS Override Settings 2]]>
                        </name>
                    </DnsOverride>
                </list>
            </dnsOverrides>
            <useRobots>IGNORE</useRobots>
            <useSitemap>false</useSitemap>
            <malwareMonitoring>false</malwareMonitoring>
```

Sample - Get details - logout regular expression list

Let us get details for the webapp with logout regular expression list.

API request

```
curl -u "USERNAME:PASSWORD" -X GET -H 'Content-type: text/xml'
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webapp/842222"
```

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/3.
0/was/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
    <WebApp>
      <id>842222</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]></url>
      <owner>
        <id>337014</id>
        <username>user_john</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Doe]]></lastName>
      </owner>
      <scope>ALL</scope>
      <attributes>
        <count>0</count>
      </attributes>
      <defaultScanner>
        <type>EXTERNAL</type>
      </defaultScanner>
      <scannerLocked>false</scannerLocked>
      <urlExcludelist>
        <count>0</count>
      </urlExcludelist>
      <urlAllowlist>
        <count>0</count>
      </urlAllowlist>
      <postDataExcludelist>
        <count>0</count>
      </postDataExcludelist>
      <logoutRegexList>
        <count>1</count>
        <list>
          <UrlEntry regex="true"><![CDATA[leave]]></UrlEntry>
        </list>
      </logoutRegexList>
      <authRecords>
        <count>0</count>
      </authRecords>
    ....
  </WebApp>
  </data>
</ServiceResponse>
```

Sample - Default authentication record details

Let us view the default authentication record details for a web application.

API request

```
curl -n -u "USERNAME:PASSWORD" -X GET -H 'Content-type: text/xml'
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webapp/53040"
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
   <count>1</count>
    <data>
        <WebApp>
            <id>53040</id>
            <name><![CDATA[WASUI-5597]]></name>
            ...
            <config>
                <defaultAuthRecord>
                    <id>9133</id>
                    <name>
                        <![CDATA[WASUI-6453]]>
                    </name>
                </defaultAuthRecord>
                  </config>
             </WebApp>
       </data>
</ServiceResponse>
```

Sample - Selenium crawl script

Let us get details for the webapp with a response that returns details of the selenium crawl script along with other details for the web application.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webapp/937657"
```

XML 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/rest/xs
d/3.0/was/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>937657</id>
            <name><![CDATA[My Web Application]]></name>
            <url><![CDATA[http://mywebapp.com]]></url>
           <owner>
                <id>337014</id>
                <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <scope>ALL</scope>
            <attributes>
                <count>0</count>
            </attributes>
            <defaultScanner>
                <type>EXTERNAL</type>
            </defaultScanner>
            <scannerLocked>false</scannerLocked>
            <urlExcludelist>
                <count>0</count>
            </urlExcludelist>
            <urlAllowlist>
                <count>0</count>
            </urlAllowlist>
            <postDataExcludelist>
                <count>0</count>
            </postDataExcludelist>
            <logoutRegexList>
                <count>0</count>
            </logoutRegexList>
            <authRecords>
                <count>0</count>
            </authRecords>
            <dnsOverrides>
                <count>0</count>
            </dnsOverrides>
            <useRobots>IGNORE</useRobots>
            <useSitemap>false</useSitemap>
            <malwareMonitoring>false</malwareMonitoring>
            <malwareNotification>false</malwareNotification>
            <tags>
                <count>0</count>
            </tags>
            <comments>
                <count>0</count>
            </comments>
            <isScheduled>false</isScheduled>
            <createdBy>
                <id>337014</id>
                <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </createdBy>
            <createdDate>2017-02-06T10:54:00Z</createdDate>
            <updatedBy>
                <id>337014</id>
                <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </updatedBy>
            <updatedDate>2017-02-06T10:54:00Z</updatedDate>
            <config/>
            <crawlingScripts>
             <count>1</count>
                <list>
                    <SeleniumScript>
                        <id>2500</id>
                        <name><![CDATA[TestSeleniumScript]]>
                        </name>
                        <data>
                            <![CDATA[
                            <?xml version="1.0" encoding="UTF-8"?>
```

```
                            <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
                            <html xmlns="http://www.w3.org/1999/xhtml"
                             xml:lang="en" lang="en">
                             <head profile="http://selenium-
ide.openqa.org/profiles/test-case">
                                    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
                                    <link rel="selenium.base"
href="http://10.10.26.238" />
                                    <title>New Test</title>
                                </head>
                                <body>
                                    <table cellpadding="1"
cellspacing="1" border="1">
                                        <thead>
                                            <tr>
                                                <td rowspan="1"
colspan="3">New Test</td>
                                            </tr>
                                        </thead>
                                        <tbody>
                                            <tr>
                                                <td>open</td>
                                                <td>http://10.10.26.23
8/</td>
                                                <td></td>
                                            </tr>
                                            <tr>
                                                <td>type</td>
                                                <td>name=login</td>
                                                <td>admin</td>
                                            </tr>
                                            <tr>
                                                <td>type</td>
                                                <td>name=password</td>
                                                <td>abc123</td>
                                            </tr>
                                            <tr>
                                                <td>clickAndWait</td>
                                                <td>name=submit</td>
                                                <td></td>
                                            </tr>
                                        </tbody>
                                    </table>
                                </body></html>]]>
                            </data>
                            <requiresAuthentication>true
                            </requiresAuthentication>
                            <startingUrl>
                                <![CDATA[http://www.mywebapp.com]]>
                            </startingUrl>
                            <startingUrlRegex>true</startingUrlRegex>
                        </SeleniumScript>
                      </list>
                    </crawlingScripts>
                </WebApp>
            </data>
        </ServiceResponse>
```

Sample - Get details of a progressive scan

If Progressive Scanning is enabled for the subscription, the progressiveScanning element is displayed in GET call responses. If Progressive Scanning is not enabled for the subscription, the element is not included. For all existing web applications created prior to WAS 4.0 the value will be set to TRUE by default.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webapp/323102"
```

XML response

```
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.
0/was/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>323102</id>
            <name>
                <![CDATA[MamboCMS]]>
            </name>
            <url>                <![CDATA[http://funkytown.abcd01.abcd
.com/Forms/FormFields/temp/updated_web_app_name]]>
```

```
            </url>
...
            <scannerLocked>false</scannerLocked>
            <progressiveScanning>DISABLED</progressiveScanning>
...
```

XSD

<platform API server>/qps/xsd/3.0/was/webapp.xsd

Create Web Application

/qps/rest/3.0/create/was/webapp

[POST]

A web application is a configuration in your account. Once created, a user can select the web application as the target of a web application scan.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access" and WAS Asset Permission "Create Web Asset". The output includes web applications in the user's scope. If you want to add postman collection files, you must have the 'ENABLE_POSTMAN_COLLECTION' option enabled for your account. If this option is not enabled, contact Qualys Support to enable this option.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements.

Click here for available operators

When only "name" and "url" are specified:

- Scope defaults to ALL. The scanner will crawl all directories and sub-directories of the starting URL.

- No default option profile is specified. An option profile must be specified for each scan.

- No authentication records are defined. No form or server authentication will be performed.

- No allows lists or exclude lists are defined. All directories and sub-directories of the starting URL will be scanned.

Samples

Create web app with minimum criteria

Create web app with one authentication record

Create web app with multiple criteria

Create web app with custom attributes

Create web app and set the default authentication record

Create web app and assign multiple scanner appliances

Create web app and add a selenium script

Create web app and configure Progressive Scanning

Create web app with non-standard TLDs

Sample - Create web app - minimum criteria

Let us create a new web application called “My Web Application” that has the starting URL “http://mywebapp.com”. The default web application settings are assigned automatically.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"<qualys_base_url>/qps/rest/3.0/create/was/webapp/" < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <data>
    <WebApp>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]></url>
    </WebApp>
  </data>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="<qualys_base_url>/qps/xsd/3.0/was/webap
p.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1912949</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]]></url>
      <owner>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <scope>ALL</scope>
      <attributes>
        <count>0</count>
      </attributes>
      <defaultScanner>
        <type>EXTERNAL</type>
      </defaultScanner>
      <scannerLocked>false</scannerLocked>
      <urlExcludelist>
        <count>0</count>
      </urlExcludelist>
      <urlAllowlist>
        <count>0</count>
      </urlAllowlist>
      <postDataExcludelist>
        <count>0</count>
      </postDataExcludelist>
      <authRecords>
        <count>0</count>
      </authRecords>
      <useRobots>IGNORE</useRobots>
      <useSitemap>false</useSitemap>
      <malwareMonitoring>false</malwareMonitoring>
      <tags>
        <count>0</count>
      </tags>
      <comments>
```

```
        <count>0</count>
      </comments>
      <isScheduled>false</isScheduled>
      <createdBy>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <createdDate>2017-10-18T18:26:40Z</createdDate>
      <updatedBy>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
      <updatedDate>2017-10-18T18:26:40Z</updatedDate>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Create web app with one authentication record

Let us create a new web application called “My Web Application” that has the starting URL “http://mywebapp.com” and has 1 authentication record.

API request

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

Request POST data

```
<ServiceRequest>
  <data>
    <WebApp>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]></url>
      <authRecords>
        <set>
          <WebAppAuthRecord>
            <id>77350</id>
          </WebAppAuthRecord>
        </set>
     </authRecords>
    </WebApp>
  </data>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1929030</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]]></url>
      <owner>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <scope>ALL</scope>
      <attributes>
        <count>0</count>
      </attributes>
      <defaultScanner>
        <type>EXTERNAL</type>
      </defaultScanner>
      <scannerLocked>false</scannerLocked>
      <urlExcludelist>
        <count>0</count>
      </urlExcludelist>
      <urlAllowlist>
        <count>0</count>
      </urlAllowlist>
      <postDataExcludelist>
        <count>0</count>
      </postDataExcludelist>
      <authRecords>
        <count>1</count>
        <list>
          <WebAppAuthRecord>
            <id>77350</id>
            <name><![CDATA[My Authentication Record]]></name>
          </WebAppAuthRecord>
        </list>
      <useRobots>IGNORE</useRobots>
     ...
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Create web app with multiple criteria

Let us create a new web application with the name “My Web Application” and the starting URL “http://www.example.com”. The web application is assigned custom settings as defined in the request POST data.

API request

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

Request POST data

```
<ServiceRequest>
     <data>
          <WebApp>
               <name><![CDATA[My Web Application]]></name>
               <url> <![CDATA[http://www.example.com]]></url>
               <scope>DOMAINS</scope>
               <domains>
<set>
<Domain><![CDATA[corp2.ab.myapp.com]]></Domain>
<Domain><![CDATA[corp1.myapp.com]]></Domain>
</set>
</domains>
     <uris>
          <set>
<Url><![CDATA[http://corp1.myapp.com]]></Url>
<Url><![CDATA[http://corp1.myapp.com/]]></Url>
<Url><![CDATA[https://corp1.myapp.com]]></Url>
```

```
<Url><![CDATA[https://corp1.myapp.com/]]></Url>
<Url><![CDATA[https://corp1.myapp.com:443]]></Url>
<Url><![CDATA[https://corp1.myapp.com:443/]]></Url>
<Url><![CDATA[http://corp1.myapp.com:8080/]]></Url>
<Url><![CDATA[http://corp1.myapp.com/startingUri]]></Url>
<Url><![CDATA[http://corp1.myapp.com/startingUri?]]></Url>
<Url><![CDATA[http://corp1.myapp.com/startingUri?param=true]]> </Url>
<Url><![CDATA[http://corp1.myapp.com/startingUri?param=true&param2=fal
se]]></Url>
<Url><![CDATA[http://corp1.myapp.com/otherUri]]></Url>
<Url><![CDATA[http://corp1.myapp.com/otherUri?param=1]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com/]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com/]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com:443]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com:443/]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com:8080/]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com/startingUri]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com/startingUri?]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com/startingUri?param=true]]></Url
>
<Url><![CDATA[http://corp2.ab.myapp.com:443/startingUri?param=true&par
am2=false]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com:8080/otherUri]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com/otherUri?param=1]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com/otherUri?param=1]]></Url>
</set>
</uris>
     <tags><set>
     <Tag><id>217118</id></Tag>
     <Tag><id>152743</id></Tag>
     <Tag><id>216368</id></Tag>
     <Tag><id>153442</id></Tag>
     </set>
     </tags>
     <defaultProfile>
          <id>90212</id>
     </defaultProfile>
     <defaultScanner>
          <type>INTERNAL</type>
          <friendlyName><![CDATA[friendlyname]]>
</friendlyName>
     </defaultScanner>
     <dnsOverrides>
```

```
          <set>
             <DnsOverride>
                <id>2022</id>
             </DnsOverride>
          </set>
     </dnsOverrides>
     <useRobots>EXCLUDELIST</useRobots>
     <useSitemap>true</useSitemap>
     <headers>
          <set>
               <WebAppHeader><![CDATA[some headers]]> </WebAppHeader>
          </set>
     </headers>
     <urlExcludelist>
          <set>
               <UrlEntry regex="true">
<![CDATA[http://rg.excludelist.*.qa.myapp.com]]></UrlEntry>
               <UrlEntry regex="true">
<![CDATA[http://rg.excludelist.*?]]></UrlEntry>
               <UrlEntry>
<![CDATA[http://url.excludelist.2.ab.myapp.com]]></UrlEntry>
               <UrlEntry regex="false">
<![CDATA[http://url.excludelist.3.qa.myapp.com]]></UrlEntry>
          </set>
        </urlExcludelist>
        <urlAllowlist>
             <set>
                  <UrlEntry regex="true">
<![CDATA[http://rg.allowlist.*.qa.myapp.com]]></UrlEntry>
                  <UrlEntry regex="true">
<![CDATA[http://rg.allowlist.*?]]></UrlEntry>
                                         <UrlEntry><![CDATA[http://url
.allowlist.2.ab.myapp.com]]></UrlEntry><UrlEntry regex="false"><![CDAT
[http://url.allowlist.3.ab.myapp.com]]></UrlEntry>
             </set>
        </urlAllowlist>
        <postDataExcludelist>
             <set>
                  <UrlEntry regex="true"><![CDATA
[http://rg.postdatexcludelist.*.ab.myapp.com]]></UrlEntry>
                  <UrlEntry
regex="true"><![CDATA[http://rg.postdatexcludelist.*?]]></UrlEntry>
             </set>
        </postDataExcludelist>
        <comments>
```

```
             <set>
                  <Comment>
                  <contents><![CDATA[some additional
comments]]></contents>
                  </Comment>
             </set>
        </comments>
        </WebApp>
   </data>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1912750</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://www.example.com]]></url>
      <owner>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <scope>DOMAINS</scope>
      <domains>
        <count>2</count>
        <list>
          <Domain><![CDATA[corp1.myapp.com]]></Domain>
          <Domain><![CDATA[corp2.ab.myapp.com]]></Domain>
        </list>
      </domains>
      <uris>
        <count>26</count>
        <list>
          <Url><![CDATA[https://corp2.ab.myapp.com]]></Url>
          <Url><![CDATA[http://corp1.myapp.com/otherUri?param=1]]></Ur
l>
          <Url><![CDATA[http://corp1.myapp.com/]]></Url>
          <Url><![CDATA[https://corp1.myapp.com]]></Url>
          <Url><![CDATA[http://corp1.myapp.com/startingUri?]]></Url>
          <Url><![CDATA[https://corp2.ab.myapp.com:443/]]></Url>
          <Url><![CDATA[https://corp2.ab.myapp.com/otherUri?param=1]]>
</Url>
          <Url><![CDATA[https://corp1.myapp.com:443/]]></Url>
          <Url><![CDATA[http://corp2.ab.myapp.com/startingUri?param=tr
ue]]></Url>
          <Url><![CDATA[http://corp2.ab.myapp.com:8080/]]></Url>
          <Url><![CDATA[http://corp1.myapp.com/otherUri]]></Url>
          <Url><![CDATA[http://corp1.myapp.com/startingUri?param=true&
param2=false]]></Url>
          <Url><![CDATA[http://corp1.myapp.com]]></Url>
          <Url><![CDATA[http://corp1.myapp.com/startingUri?param=true]
]></Url>
          <Url><![CDATA[http://corp2.ab.myapp.com]]></Url>
          <Url><![CDATA[https://corp2.ab.myapp.com/]]></Url>
          <Url><![CDATA[http://corp2.ab.myapp.com/]]></Url>
          <Url><![CDATA[https://corp2.ab.myapp.com:443]]></Url>
          <Url><![CDATA[http://corp1.myapp.com/startingUri]]></Url>
          <Url><![CDATA[https://corp1.myapp.com:443]]></Url>
          <Url><![CDATA[http://corp2.ab.myapp.com/startingUri]]></Url>
          <Url><![CDATA[http://corp1.myapp.com:8080/]]></Url>
          <Url><![CDATA[https://corp2.ab.myapp.com:8080/otherUri]]></U
rl>
          <Url><![CDATA[https://corp1.myapp.com/]]></Url>
          <Url><![CDATA[http://corp2.ab.myapp.com/startingUri?]]></Url
>        <Url><![CDATA[http://corp2.ab.myapp.com:443/startingUri?param
=true&param2=false]]></Url>
        </list>
      </uris>
      <defaultProfile>
        <id>90212</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </defaultProfile>
      <defaultScanner>
        <type>INTERNAL</type>
        <friendlyName><![CDATA[friendlyname]]></friendlyName>
      </defaultScanner>
      <scannerLocked>false</scannerLocked>
      <dnsOverrides>
           <set>
              <DnsOverride>
                  <id>2022</id>
             </DnsOverride>
           </set>
      </dnsOverrides>
      <urlExcludelist>
        <count>4</count>
        <list>
          <UrlEntry
regex="false"><![CDATA[http://url.excludelist.2.ab.myapp.com]]></UrlEn
try>
          <UrlEntry
regex="false"><![CDATA[http://url.excludelist.3.ab.myapp.com]]></UrlEn
try>
          <UrlEntry
regex="true"><![CDATA[http://rg.excludelist.*.ab.myapp.com]]></UrlEntr
y>
          <UrlEntry
regex="true"><![CDATA[http://rg.excludelist.*?]]></UrlEntry>
        </list>
      </urlExcludelist>
      <urlAllowlist>
        <count>4</count>
        <list>
          <UrlEntry
regex="true"><![CDATA[http://rg.allowlist.*.ab.myapp.com]]></UrlEntry>
          <UrlEntry
regex="true"><![CDATA[http://rg.allowlist.*?]]></UrlEntry>
          <UrlEntry
regex="false"><![CDATA[http://url.allowlist.2.ab.myapp.com]]></UrlEntr
y>
          <UrlEntry
regex="false"><![CDATA[http://url.allowlist.3.ab.myapp.com]]></UrlEntr
y>
        </list>
      </urlAllowlist>
      <postDataExcludelist>
        <count>2</count>
        <list>
          <UrlEntry
regex="true"><![CDATA[http://rg.postdataexcludelist.*.ab.myapp.com]]></
UrlEntry>
          <UrlEntry
regex="true"><![CDATA[http://rg.postdataexcludelist.*?]]></UrlEntry>
        </list>
      </postDataExcludelist>
      <authRecords>
        <count>0</count>
      </authRecords>
      <useRobots>EXCLUDELIST</useRobots>
      <useSitemap>true</useSitemap>
      <headers>
        <count>1</count>
        <list>
          <WebAppHeader><![CDATA[some headers]]></WebAppHeader>
        </list>
      </headers>
      <malwareMonitoring>false</malwareMonitoring>
      <tags>
        <count>4</count>
        <list>
          <Tag>
            <id>152743</id>
            <name><![CDATA[Asset Groups]]></name>
          </Tag>
          <Tag>
            <id>217118</id>
            <name><![CDATA[AUG 27]]></name>
          </Tag>
          <Tag>
            <id>153442</id>
            <name><![CDATA[Malware Domain Assets]]></name>
          </Tag>
          <Tag>
            <id>216368</id>
            <name><![CDATA[Asset name rule]]></name>
          </Tag>
        </list>
      </tags>
      <comments>
        <count>1</count>
        <list>
          <Comment>
            <contents><![CDATA[some additional comments]]></contents>
            <createdDate>2017-10-18T17:57:32Z</createdDate>
          </Comment>
        </list>
      </comments>
      <isScheduled>false</isScheduled>
      <createdBy>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <createdDate>2017-10-18T17:57:32Z</createdDate>
      <updatedBy>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
      <updatedDate>2017-10-18T17:57:32Z</updatedDate>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Create web app with custom attributes

API request

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

Request POST data

```
<ServiceRequest>
  <data>
    <WebApp>
        <name><![CDATA[Custom Attribute via API]]></name>
<url><![CDATA[http://funkytown.vuln.qa.qualys.com:80/updated_web_app_n
ame/]]></url>
        <attributes>
            <set>
            <Attribute>
             <name>Custom key 1</name>
             <value><![CDATA[Custom value 1]]></value>
            </Attribute>
            </set>
        </attributes>
    </WebApp>
   </data>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>2514680</id>
      <name><![CDATA[Custom Attribute via API]]></name>
<url><![CDATA[http://funkytown.vuln.qa.qualys.com:80/updated_web_app_n
ame/]]></url>
      <owner>
        <id>4354</id>
        <username>user_steve</username>
        <firstName><![CDATA[Steve]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <scope>ALL</scope>
      <attributes>
        <count>1</count>
        <list>
          <Attribute>
            <name><![CDATA[Custom key 1]]></name>
            <value><![CDATA[Custom value 1]]></value>
          </Attribute>
        </list>
      </attributes>
      <defaultScanner>
        <type>EXTERNAL</type>
      </defaultScanner>
      <scannerLocked>false</scannerLocked>
      <progressiveScanning>ENABLED</progressiveScanning>
      <urlExcludelist>
        <count>0</count>
      </urlExcludelist>
      <urlAllowlist>
        <count>0</count>
      </urlAllowlist>
      <postDataExcludelist>
        <count>0</count>
      </postDataExcludelist>
      <authRecords>
        <count>0</count>
      </authRecords>
      <dnsOverrides>
        <count>0</count>
      </dnsOverrides>
      <useRobots>IGNORE</useRobots>
      <useSitemap>false</useSitemap>
      <malwareMonitoring>false</malwareMonitoring>
      <tags>
        <count>0</count>
      </tags>
      <comments>
        <count>0</count>
      </comments>
      <isScheduled>false</isScheduled>
      <createdBy>
        <id>4354</id>
        <username>user_steve</username>
        <firstName><![CDATA[Steve]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <createdDate>2017-09-30T00:18:38Z</createdDate>
      <updatedBy>
        <id>4354</id>
        <username>user_steve</username>
        <firstName><![CDATA[Steve]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
      <updatedDate>2017-09-30T00:18:38Z</updatedDate>
      <config/>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Create web app and set the default authentication record

Let us configure the default authentication record while creating or updating the web application. Create a web application with default authentication record ID #9133.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
```

```
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/webapp/" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <data>
        <WebApp>
            <name>
               <![CDATA[Create webapp with default auth record]]>
            </name>
            <url><![CDATA[http://mywebapp.com]]></url>
            <scope>ALL</scope>
            <scannerLocked>false</scannerLocked>
            <useRobots>IGNORE</useRobots>
            <useSitemap>false</useSitemap>
            <malwareMonitoring>false</malwareMonitoring>
            <config>
                <defaultAuthRecord>
                    <id>9133</id>
                </defaultAuthRecord>
            </config>
         </WebApp>
    </data>
</ServiceRequest>
```

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/3.
0/
was/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>53040</id>
            <name>
               <![CDATA[Create webapp with default auth record]]>
            </name>
            ...
            <config>
                <defaultAuthRecord>
```

```
                    <id>9133</id>
                    <name>
                        <![CDATA[WAS-9133]]>
                    </name>
                </defaultAuthRecord>
            </config>
        </WebApp>
</data>
</ServiceResponse>
```

Sample - Create web app and assign multiple scanner appliances

Let us create a new web application called “My Web Application” with the starting URL “http://mywebapp.com” and assign a group of scanners using tag Scannerpool (ID 15415353311147). The default web application settings are assigned automatically.

API request

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

Request POST data

```
<ServiceRequest>
   <data>
      <WebApp>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]></url>
      <defaultScannerTags>
          <set>
            <Tag>
              <id>15415353311147</id>
            </Tag>
          </set>
      </defaultScannerTags>
     </WebApp>
   </data>
</ServiceRequest>
```

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/3.
0/
was/webapp.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
<data>
    <WebApp>
      <id>842422</id>
      <name><![CDATA[My Web Application]></name>
      <url><![CDATA[http://mywebapp.com]]></url>
      <owner>
        <id>337014</id>
        <username>user_john</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Doe]]></lastName>
      </owner>
      <scope>ALL</scope>
      <attributes>
        <count>0</count>
      </attributes>
      <defaultScannerTags>
        <count>1</count>
          <list>
             <Tag>
               <id>15415353311147</id>
               <name>
                 <![CDATA[TagForScanner]]>
               </name>
             </Tag>
          </list>
      </defaultScannerTags>
<scannerLocked>false</scannerLocked>
<progressiveScanning>DISABLED</progressiveScanning>
      <urlExcludelist>
        <count>0</count>
      </urlExcludelist>
      <urlAllowlist>
        <count>0</count>
      </urlAllowlist>
      <postDataExcludelist>
        <count>0</count>
      </postDataExcludelist>
      <logoutRegexList>
```

```
        <count>0</count>
      </logoutRegexList>
      <authRecords>
        <count>0</count>
      </authRecords>
      <dnsOverrides>
        <count>0</count>
      </dnsOverrides>
      <useRobots>IGNORE</useRobots>
      <useSitemap>false</useSitemap>
      <malwareMonitoring>false</malwareMonitoring>
      <tags>
        <count>0</count>
      </tags>
      <comments>
        <count>0</count>
      </comments>
      <isScheduled>false</isScheduled>
      <createdBy>
        <id>337014</id>
        <username>user_john</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Doe]]></lastName>
      </createdBy>
      <createdDate>2017-01-12T12:03:37Z</createdDate>
      <updatedBy>
        <id>337014</id>
        <username>user_john</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Doe]]></lastName>
      </updatedBy>
      <updatedDate>2017-01-12T12:03:37Z</updatedDate>
      <config/>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Create web app and add a selenium script

Let us create a new web application called “My Web Application” that has the starting URL “http://mywebapp.com” and add selenium script (TestSeleniumScript) to it.

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <WebApp>
            <name><![CDATA[My Web Application]]></name>
            <url><![CDATA[http://mywebapp.com]]></url>
            <crawlingScripts>
                <set>
                        <SeleniumScript>
                        <name><![CDATA[TestSeleniumScript]]></name>
                        <startingUrl><![CDATA[http://www.mywebapp.com]
]>
                        </startingUrl>
                        <data>
                            <![CDATA[<?xml version="1.0"
encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://10.10.26.238" /><title>New
Test</title></head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">New Test</td></tr>
</thead><tbody><tr><td>open</td><td>http://10.10.26.238/</td><td></td>
</tr><tr><td>type</td><td>name=login</td><td>admin</td></tr><tr><td>ty
pe</td><td>name=password</td><td>abc123</td></tr><tr><td>clickAndWait<
/td><td>name=submit</td><td></td></tr></tbody></table></body></html>]]
>
                        </data>
                        <requiresAuthentication>true</requiresAuthenti
cation>
                        <startingUrlRegex>true</startingUrlRegex>
                    </SeleniumScript>
                </set>
            </crawlingScripts>
        </WebApp>
    </data>
</ServiceRequest>
```

XML 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/rest/xs
d/3.0/was/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>937657</id>
            <name><![CDATA[My Web Application]]></name>
            <url><![CDATA[http://mywebapp.com]]></url>
            <owner>
                <id>337014</id>
                <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <scope>ALL</scope>
            <attributes>
                <count>0</count>
            </attributes>
            <defaultScanner>
                <type>EXTERNAL</type>
            </defaultScanner>
            <scannerLocked>false</scannerLocked>
            <urlExcludelist>
                <count>0</count>
            </urlExcludelist>
            <urlAllowlist>
                <count>0</count>
            </urlAllowlist>
            <postDataExcludelist>
                <count>0</count>
            </postDataExcludelist>
            <logoutRegexList>
                <count>0</count>
            </logoutRegexList>
            <authRecords>
```

```
                <count>0</count>
            </authRecords>
            <dnsOverrides>
                <count>0</count>
            </dnsOverrides>
            <useRobots>IGNORE</useRobots>
            <useSitemap>false</useSitemap>
            <malwareMonitoring>false</malwareMonitoring>
            <malwareNotification>false</malwareNotification>
            <tags>
                <count>0</count>
            </tags>
            <comments>
                <count>0</count>
            </comments>
            <isScheduled>false</isScheduled>
            <createdBy>
                <id>337014</id>
                <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </createdBy>
            <createdDate>2017-02-06T10:54:00Z</createdDate>
            <updatedBy>
                <id>337014</id>
                <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </updatedBy>
            <updatedDate>2017-02-06T10:54:00Z</updatedDate>
            <config/>
            <crawlingScripts>
                <count>1</count>
                <list>
                    <SeleniumScript>
                        <id>2500</id>
                        <name>
                            <![CDATA[TestSeleniumScript]]>
                        </name>
                        <data>
                            <![CDATA[
                            <?xml version="1.0" encoding="UTF-8"?>
                            <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
                            <html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
                                <head profile="http://selenium-
ide.openqa.org/profiles/test-case">
                                    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
                                    <link rel="selenium.base"
href="http://10.10.26.238" />
                                    <title>New Test</title>
                                </head>
                                <body>
                                    <table cellpadding="1"
cellspacing="1" border="1">
                                        <thead>
                                            <tr>
                                                <td rowspan="1"
colspan="3">New Test</td>
                                            </tr>
                                        </thead>
                                        <tbody>
                                            <tr>
                                                <td>open</td>
                                                <td>http://10.10.26.23
8/</td>
                                                <td></td>
                                            </tr>
                                            <tr>
                                                <td>type</td>
                                                <td>name=login</td>
                                                <td>admin</td>
                                            </tr>
                                            <tr>
                                                <td>type</td>
                                                <td>name=password</td>
                                                <td>abc123</td>
                                            </tr>
                                            <tr>
                                                <td>clickAndWait</td>
                                                <td>name=submit</td>
                                                <td></td>
                                            </tr>
                                        </tbody>
                                    </table>
                                </body></html>]]>
                            </data>
                            <requiresAuthentication>true
                            </requiresAuthentication>
                            <startingUrl>
                                <![CDATA[http://www.mywebapp.com]]>
                            </startingUrl>
                            <startingUrlRegex>true</startingUrlRegex>
                        </SeleniumScript>
                       </list>
                    </crawlingScripts>
                </WebApp>
            </data>
        </ServiceResponse>
```

Sample: Progressive Scanning

The user will be able to set progressiveScanning to true or false, if Progressive Scanning is enabled for the subscription. When Progressive Scanning is enabled for the subscription, if progressiveScanning option is not specified during CREATE request, by default the option will be true for the web application.

API request

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

Request POST data

```
<ServiceRequest>
    <data>
    <WebApp>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]></url>
      <progressiveScanning>false</progressiveScanning>
    </WebApp>
  </data>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1912949</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[http://mywebapp.com]]]></url>
...
            <scannerLocked>false</scannerLocked>
            <progressiveScanning>false</progressiveScanning>
...
```

If Progressive Scanning is not enabled for the subscription, the <progressiveScanning> element cannot not be provided, otherwise an error will be returned.

XML response (error)

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.
0/was/webapp.xsd">
    <responseCode>INVALID_REQUEST</responseCode>
    <responseErrorDetails>
        <errorMessage>Progressive scanning is not enabled in your
subscription.</errorMessage>
        <errorResolution>Please check with your account manager to
enable this option.</errorResolution>
    </responseErrorDetails>
</ServiceResponse>
```

Sample: Create web app with non-standard TLDs

The user will be able to create a new web application by adding URLs with non-standard TLDs in domain.

API request

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

Request POST data

```
<ServiceRequest>
   <data>
      <WebApp>
         <name><![CDATA[TEST_TLDs_latest-1 - API]]></name>
         <url><![CDATA[http://10.11.68.74]]></url>
         <scope>DOMAINS</scope>
         <domains>
            <set>
               <Domain><![CDATA[Afterfix.showtime]]></Domain>
            </set>
         </domains>
      </WebApp>
   </data>
</ServiceRequest>
```

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/3.0/was/webapp.xsd">
   <responseCode>SUCCESS</responseCode>
   <count>1</count>
   <data>
      <WebApp>
         <id>11713839</id>
         <name><![CDATA[TEST_TLDs_latest-1 - API]]></name>
         <url><![CDATA[http://10.11.68.74]]></url>
         <owner>
            <id>8296038</id>
            <username>user_john</username>
            <firstName><![CDATA[John]]></firstName>
            <lastName><![CDATA[Doe]]></lastName>
         </owner>
         <scope>DOMAINS</scope>
         ....
      </WebApp>
   </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/webapp.xsd

Update Web Application

/qps/rest/3.0/update/was/webapp/<id>

[POST]

Update a web application configuration in your account.

Note: The user can add URLs with non-standard TLDs in domain while updating a web application

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and WAS Asset Permission “Edit Web Asset”, "Edit Web Application URL" and "Select and Lock/Unlock Scanner Appliance". The output includes web applications in the user's scope. If you want to add postman collection files, you must have the 'ENABLE_POSTMAN_COLLECTION' option enabled for your account. If this option is not enabled, contact Qualys Support to enable this option.

Input Parameters

The element “id” (integer) is required, where “id” identifies a web application.

Click here for available operators

Samples

Update web app with minimum information

Update authentication records for web app

Update multiple settings

Update web app to set default cancel time

Update custom attribute value for the web app

Update the default authentication record of the web app

Sample - Update web app with minimum information

Let us update information for the web application with ID 1234, change the name to “My WebApp Name” .

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/1234" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <data>
    <WebApp>
      <name>My WebApp Name</name>
    </WebApp>
  </data>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1234</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Update authentication records for web app

Let us update web application with ID 1234, add 1 authentication record and remove 1 authentication record.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/1234" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <data>
    <WebApp>
       <name><![CDATA[My WebApp Name]]></name>
      <authRecords>
        <add>
          <WebAppAuthRecord>
            <id>77355</id>
          </WebAppAuthRecord>
        </add>
        <remove>
          <WebAppAuthRecord>
            <id>77356</id>
          </WebAppAuthRecord>
        </remove>
      </authRecords>
    </WebApp>
  </data>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>1234</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Update multiple settings

Let us update multiple settings for a web application. The web application is assigned custom settings as defined in the request POST data.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/2607056"
< file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
     <data>
          <WebApp>
               <name>My Web Application</name>
      <url>http://mywebapp.com</url>
      <attributes>
        <remove>
          <Attribute>
            <name>Business Function</name>
           </Attribute>
          <Attribute>
            <name>Business Location</name>
          </Attribute>
        </remove>
        <update>
          <Attribute>
            <name>Business Description</name>
            <value>Business Description Value - UPDATED</value>
          </Attribute>
        </update>
      </attributes>
      <defaultProfile><id>365333</id></defaultProfile>
      <urlExcludelist>
        <set>     <UrlEntry><![CDATA[http://url.excludelist.1.mywebapp
.com]]></UrlEntry>
```

```
          <UrlEntry
regex="false"><![CDATA[http://url.excludelist.2.mywebapp.com]]></UrlEn
try>
          <UrlEntry
regex="true"><![CDATA[http://rg.excludelist.*.com]]></UrlEntry>
        </set>
      </urlExcludelist>
      <urlAllowlist>
        <set>
<UrlEntry><![CDATA[http://url.allowlist.1.mywebapp.com]]></UrlEntry>
        <UrlEntry
regex="false"><![CDATA[http://url.allowlist.2.mywebapp.com]]></UrlEntr
y>
        <UrlEntry
regex="true"><![CDATA[http://rg.allowlist.*.mywebapp.com]]></UrlEntry>
        </set>
      </urlAllowlist>
      <postDataExcludelist>
        <set>
        <UrlEntry
regex="true"><![CDATA[http://url.postdataexcludelist.1.mywebapp.com]]>
</UrlEntry>
        <UrlEntry
regex="true"><![CDATA[http://url.postdataexcludelist.2.mywebapp.com]]>
</UrlEntry>
        </set>
      </postDataExcludelist>
      <useRobots>ADD_PATHS</useRobots>
      <useSitemap>true</useSitemap>
      <headers>
        <set>
          <WebAppHeader>X-TTP-REQUESTED-BY: Qualys Test</WebAppHeader>
        </set>
      </headers>
    </WebApp>
   </data>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>2607056</id>
        </WebApp>
    </data>
</ServiceResponse>
```

Sample - Update web app to set default cancel time

Let us set the default cancel scan option for web application ID 2392272. Scans of this web application will be set to cancel at 10pm by default.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/2392272"
< file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <data>
    <WebApp>
    <name><![CDATA[My Web App]]></name>
    <url><![CDATA[http://mywebapp.com]]></url>
    <config><cancelScansAt>22:00</cancelScansAt></config>
  </WebApp>
   </data>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.qualys.com/qps
/xsd/3.0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebApp>
      <id>2392272</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Update custom attribute value for the web app

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/2514679"
< file.xml
Note: "file.xml" contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <data>
        <WebApp>
            <attributes>
                <update>
                    <Attribute>
                     <name>Custom key 1</name>
                     <value><![CDATA[Custom value 1]]></value>
                    </Attribute>
                </update>
            </attributes>
         </WebApp>
    </data>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>2514679</id>
        </WebApp>
    </data>
</ServiceResponse>
```

Sample - Update the default authentication record of the web app

Let us update the default authentication record for the web application with ID 33831.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webapp/33831" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <data>
        <WebApp>
            <config>
             <defaultAuthRecord>
                  <id>9133</id>
             </defaultAuthRecord>
          </config>
        </WebApp>
    </data>
</ServiceRequest>
```

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/3.
0/
was/webapp.xsd">
<responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebApp>
            <id>33831</id>
        </WebApp>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/webapp.xsd

Delete Web Application

/qps/rest/3.0/delete/was/webapp/<id>

/qps/rest/3.0/delete/was/webapp/<filters>

[POST]

Delete a web application configuration in your account.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and WAS Asset Permission “Delete Web Asset”. The web application to be deleted must be within the user’s scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements.

Click here for available operators

Parameter	Description
id	(integer) Web application ID.
name	(text) Web application name.
url	(text) The URL of web application.
removeFromSubscription	(Boolean) When set to true, deletes the web application asset from your subscription if the web application is not shared with other modules such as WAF. The “removeFromSubscription” flag is ignored if the web application that you want to remove from the subscription is shared with other modules. In that case, the Delete Web

	application API request with this flag set to true will only delete the web application from WAS and not from your subscription.
tags.name	(text) Tag name assigned to web application.
tags.id	(integer) Tag ID assigned to web application.
createdDate	(date) The date when the web application was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the web application was last updated in WAS, in UTC date/time format.
isScheduled	(boolean) A flag indicating whether a scan is scheduled for web application.
isScanned	(boolean) A flag indicating whether the web application has been scanned.
lastScan.status	(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED
lastScan.date	(date) Date when web application was last scanned, in UTC date/time format.

Sample - Delete a single web application

Let us delete the web application that has the ID 1234.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X
POST" "https://qualysapi.qualys.com/qps/rest/3.0/delete/was/webapp/123
4"
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
   <count>1</count>
    <data>
      <WebApp>
        <id>1234</id>
      </WebApp>
    </data>
</ServiceResponse>
```

Sample - Delete bulk web applications

Let us delete web applications in the user’s account that have a name with the word “Merchant” and have an ID greater than 323000.

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">Merchant</Criteria>
    <Criteria field="id" operator="GREATER">323000</Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
  <data>
    <WebApp>
      <id>323126</id>
    </WebApp>
    <WebApp>
      <id>324256</id>
    </WebApp>
    <WebApp>
      <id>323476</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Delete multiple web applications and remove the web applications from subscription

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"<qualys_base_url>/qps/rest/3.0/delete/was/webapp?action=removeFromSub
scription" < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">New
Webapp</Criteria>
        <Criteria field="id" operator="LESSER">28297453</Criteria>
    </filters>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
   <ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="
    <qualys_base_url>/qps/xsd/3.0/was/webapp.xsd">
        <responseCode>SUCCESS</responseCode>
        <count>2</count>
        <data>
            <WebApp>
```

```
                <id>28297451</id>
            </WebApp>
            <WebApp>
                <id>28297452</id>
            </WebApp>
        </data>
    </ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/webapp.xsd

Purge Web Application

/qps/rest/3.0/purge/was/webapp/<id>

/qps/rest/3.0/purge/was/webapp/<filters>

[POST]

Purging a web application results in removal of the scan findings from the web application's scan history. Henceforth, the newly generated web application reports will not include findings from previously completed scans. All dates must be entered in UTC date/time format.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and WAS Asset Permission “Purge Web Asset”. The web application to be purged must be within the user’s scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements.

Click here for available operators

Parameter	Description
id	(integer) Web application ID.
name	(text) Web application name.
url	(text) The URL of web application.
tags.name	(text) Tag name assigned to web application.
tags.id	(integer) Tag ID assigned to web application.

createdDate	(date) The date when the web application was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the web application was last updated in WAS, in UTC date/time format.
isScheduled	(boolean) A flag indicating whether a scan is scheduled for web application.
isScanned	(boolean) A flag indicating whether the web application has been scanned.
lastScan.status	(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED
lastScan.date	(date) Date when web application was last scanned, in UTC date/time format.

Sample - Purge a single web application

Let us purge the web application with ID 32420.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X
"POST"  https://qualysapi.qualys.com/qps/rest/3.0/purge/was/webapp/324
20"
```

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/3.
0/was/webapp.xsd">
 <responseCode>SUCCESS</responseCode>
 <count>1</count>
  <data>
    <WebApp>
      <id>32420</id>
    </WebApp>
  </data>
</ServiceResponse>
```

Sample - Purge multiple web applications

Let us purge web applications in the user’s account that have a name with the word “Merchant” and have an ID greater than 323000.

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">Merchant</Criteria>
    <Criteria field="id" operator="GREATER">323000</Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/was/webapp.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
  <data>
    <WebApp>
      <id>323126</id>
    </WebApp>
    <WebApp>
      <id>324256</id>
    </WebApp>
    <WebApp>
      <id>323476</id>
    </WebApp>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/webapp.xsd

Download Selenium Script

/qps/rest/3.0/downloadSeleniumScript/was/webapp

[POST]

Download the selenium script file that is associated with the web application.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and WAS Asset Permission “View/download Selenium Script sensitive contents”. The web application to be purged must be within the user’s scope.

Input Parameters

The element “id” (integer) is required, where “id” identifies a web application.

Click here for available operators

Sample - Download selenium script

Let us download the selenium script file associated with a web application with ID 1234.

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
        <Criteria field="id" operator="EQUALS">1234</Criteria>
        <Criteria field="crawlingScripts.id"
operator="EQUALS">2500</Criteria>
    </filters>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
    <head profile="http://selenium-ide.openqa.org/profiles/test-case">
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
        <link rel="selenium.base" href="http://10.10.26.238" />
        <title>New Test</title>
    </head>
    <body>
        <table cellpadding="1" cellspacing="1" border="1">
            <thead>
                <tr>
                    <td rowspan="1" colspan="3">New Test</td>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>open</td>
                    <td>http://10.10.26.238/</td>
                    <td/>
                </tr>
                <tr>
                    <td>type</td>
                    <td>name=login</td>
                    <td>admin</td>
                </tr>
                <tr>
                    <td>type</td>
                    <td>name=password</td>
                    <td>abc123</td>
                </tr>
                <tr>
                    <td>clickAndWait</td>
                    <td>name=submit</td>
                    <td/>
                </tr>
            </tbody>
        </table>
    </body>
</html>
```

XSD

<platform API server>/qps/xsd/3.0/was/webapp.xsd

Reference: WebApp

The <WebApp> element includes sub elements used to define a web application. A reference of these elements is provided below. An asterisk * indicates a complex element.

Parameter	Description
id	(integer) Web application ID. This element is assigned by the service and required for an update request.
removeFromSubscription	(Boolean) When set to true, deletes the web application asset from your subscription if the web application is not shared with other modules such as WAF. The “removeFromSubscription” flag is ignored if the web application that you want to remove from the subscription is shared with other modules. In that case, the Delete Web application API request with this flag set to true will only delete the web application from WAS and not from your subscription.
reactivateIfExists	(Boolean) Set this parameter to “true” to create a web application with the same name and URL. In such a case, all the data of the old web application such as findings, detections, scans will be deleted. The new web application will have the same web application asset ID as the old web application. But if you try to create a web application with different URL but with a name that already exists in your subscription, then the API will return an error “Webapp with same name exists” in the response. The flag "reactivateIfExists" will be ignored even if it is set to true". If this flag is not set to true and if you try to create a web application with the same name and URL, then we show this error message in the response: “We found in your subscription an existing asset that already uses the same name and URL. The

	asset is currently being used by the modules: Was, Waf. Please set flag reactivateIfExists to true to use that existing asset. If not, you will need to change the name of the one you are trying to create.”
name	(text) The web application name (maximum 256 characters). This element is required to create a web application.
type	(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.
url	(text) The URL of the web application maximum 2048 characters). This element is required to create a web application.
os	(text) The operating system of the web application.
owner	(text) This element is assigned by the service and may be specified for an update request only.
config*	Configure the cancel scan option. Specify “cancel after” time or “cancel at” time. Only one of <cancelScansAfterNHours> or <cancelScanstAt> is allowed in one config section. Example for “cancel after” time: <config> <cancelScansAfterNHours>3 </cancelScansAfterNHours> </config> Example for “cancel at” time: <config> <cancelScansAt>2017-06-10T12:00:00Z

```
</cancelScansAt>
</config>
```

Notes about updating web applications:
- If none of the above elements are specified in the config section, the default cancel option is removed from the web app settings.

- If the config section is not specified, no changes are made to the web app settings.

You can set one of the DNS override records that you assigned to your web application as the default record for the web application. The default DNS override setting is useful when you want to scan multiple web applications using the DNS override option. We will use the default DNS override record that you have set for your web applications to launch scan on them.

The parameter for setting the default DNS override is config.defaultDnsOverride.id. This parameter takes the ID of the DNS override record that you want to set as the default record.

This is an optional parameter.

Example:

```
<config>
  <defaultDnsOverride>
    <id>14620</id>
  <defaultDnsOverride>
</config>
```

attributes*

Custom web application attributes.

Example:

```
<attributes>
 <set>
   <Attribute>
     <name>Custom key 1</name>
```

	<value><![CDATA[Custom value 1]]></value> </Attribute> <Attribute> <name>Custom key 2</category> <value><![CDATA[Custom value 2]]></value> </Attribute> </set> </attributes>
tags*	Tags assigned to the web application. Example: <tags> <set> <Tag> <id>12345</id> </Tag> <Tag> <id>12345678</id> </Tag> </set> </tags>
comments	(text) Comments on the web application.
scope	(keyword) The scanning scope for the web application: ALL (default), LIMIT, SUBDOMAIN or DOMAINS. - If set to ALL, the scan will crawl all directories and sub-directories of the starting URL. - If set to LIMIT, crawling will be limited to the starting URI's initial path and sub-directories. - If set to SUBDOMAINS, any sub-domain that is in the same domain as the specified domain name will be crawled. - If set to DOMAINS, only the specified domains will be crawled.

uris	(text) Additional URLs to crawl. Each must be a valid HTTP or HTTPS URL consistent with the web application scope.

swaggerFile	Swagger-based REST API file that you want to scan for vulnerabilities. To scan the API, you need to specify the content of the Swagger/OpenAPI file in YAML or JSON format. Note that we support scanning single API at a time. For scanning Swagger-based REST APIs, the web application URL should point to the Swagger file host or OpenAPI server URL as per the API definition. Before adding the file content, you must encode the file content into base64 format. It is your responsibility to verify that you have permission to scan APIs that you specify as scan targets.
	
	To remove the API file that you added to the web application, add a blank “swaggerFile” tag in the update web application request.
	
	We currently only support Swagger API file version 2.0 and 3.0 in YAML or JSON format. The size of the file you upload should not exceed 5 MB.
	
	Example:
	
	`<WebApp>`
	  `<id>87452</id>`
	  …
	  `<swaggerFile>`
	    `<name>ajax.yml</name>`
	    `<content>LS0tDQpzd2FnZ2`
	    `VyOiAnMi4wJw0KaW5mbzoN…</content>`
	   `</swaggerFile>`
	
	Note that the swaggerFile and postmanCollection tags are mutually exclusive and cannot be specified together in the request.

postmanCollection	Postman collection files that you want to scan for vulnerabilities. postmanCollection has 3 tags for specifying Postman Collection File content:

“collection” for specifying Postman Collection File content, “environmentVariable” for specifying Postman Environment Variables File, and “globalVariable” for specifying Global Variables File. All these 3 tags are part of the “postmanCollection” tag. While creating the web application, the Postman Collection File is a mandatory parameter whereas specifying the Postman Environmental Variables and Postman Global Variables files is optional.

Note that before adding the file content, you must encode the file content into base64 format.

You can remove the files by sending blank tags in the update request. To remove,

- Postman Environment Variables File, send a blank “environmentVariable” tag.

- Postman Global Variables File, send a blank “globalVariable” tag.

- Postman Collection File, send either a blank "postmanCollection" or “collection” tag. This will also remove the variables file if added.

We currently only support v2.0.0 and v2.1.0. for Postman Collection. The size of the file you upload should not exceed 5 MB.

```
<WebApp>
<id>87452</id>
...
<postmanCollection>
  <collection>
   <name>Mycollection.json</name>
   <content>ewoJInZhcmlhYmx
   lcydLAoJImlu...</content>
  </collection>
 <environmentVariable>
   <name>Myenvvariables</name>
   <content>ewoJImlkIjogIjcxN
```

	TBhYjIyLWE1MDQtNGEz...</content> </environmentVariable> <globalVariable> <name>myglobal.json</name> <content>ewogICJpZIwNTY5Yzkz YS02YzRjLWFkMDIt...</content> </globalVariable> </postmanCollection> Note that the swaggerFile and postmanCollection tags are mutually exclusive and cannot be specified together in the request..
malwareMonitoring	(boolean) A flag indicating whether Malware Monitoring is enabled for the web application. Example:<malwareMonitoring>true</malwareMonitoring>
malwareNotification	(boolean) A flag indicating whether email notification is enabled for Malware Monitoring scans. Example:<malwareNotification>true</malwareNotification>
malwareScheduling*	Schedule Malware Monitoring scans for your web application with various scheduling options. <occurrenceType> can be set to one of: ONCE, HOURLY, DAILY, WEEKLY, MONTHLY.
Scan Settings	
defaultProfile*	The default option profile for scanning the web application. When unspecified, an option profile must be specified by the user for each scan. <defaultProfile> <id>139359</id>

	<name><![CDATA[10 Links edit]]></name> </defaultProfile>
defaultScanner*	The default scanner for the web application. A default scanner is optional. For type (keyword) specify INTERNAL for a scanner appliance. If type is INTERNAL, specify friendlyName (text). EXTERNAL for the external scanners or scannerTags for assigning multiple scanner appliances grouped by asset tag. Example: <defaultScanner> <type>INTERNAL</type> <friendlyName>dp_scanner</friendlyName> </defaultScanner>
proxy.id	(integer) The default proxy for scanning the web application. Example: <proxy> <id>12345</id> </proxy>
scannerLocked	(boolean) A flag indicating whether the default scanner appliance is locked for the web application. Example: <scannerLocked>false</scannerLocked>
dnsOverrides*	Assign DNS override settings, one or more records, to a web application.

	Example: <dnsOverrides> <set> <DnsOverride> <id>2022</id> </DnsOverride> </set> </dnsOverrides>
useRobots (keyword)	A flag indicating whether to observe the Robots.txt file and its directives if found when scanning the web application. If set to IGNORE (default) the Robots.txt file is ignANDed. If set to ADD_PATHS, the “disallow” and “allow” directives in the Robots.txt file will be observed; this means these directives will be added as link hints for the crawler. If set to EXCLUDELIST the “disallow” directives in the Robots.txt file will be observed; this means scans will not crawl matching links.
useSitemap (Boolean)	A flag indicating whether to adhere to a sitemap.xml file if present in the web application: true or false (default).
headers*	The headers that need to be injected by the scanning engine to scan the web application for complex authentication schemes or to impersonate a web browser.
urlExcludelist*	The URLs for the exclude list. These are web application links (URLs) that you do not want scanned. For each URL, specify UrlEntry (text). If the attribute regex (Boolean) is set to “true” the service performs a regular expression match.

urlAllowlist*	The URLs for the allow list. These are web application links (URLs) that you want to be scanned. For each URL, specify UrlEntry (text). If the attribute regex (Boolean) is set to "true" the service performs a regular expression match.
postDataExcludelist*	The web application URLs for which you want to prevent form submission (POST data), as this could have unwanted side effects. For each URL, specify UrlEntry (text). The attribute regex (Boolean) can be set to “true” for a regular expression match.
authRecords*	The web application authentication records. The WebAppAuthRecords element identifies a set of authentication instances (combination of form and types).
WebAppAuthRecord *	Under <authRecords>, this element identifies an authentication record assigned to the web application. Prior to WAS 3.1, authentication records and their settings were defined here using the Web Applicatin API. Now you can manage authentication records using the Authentication API.
CrawlingScript	The selenium crawl script for your web application. The SeleniumScript element tells the selenium script details.
SeleniumScript	Under <CrawlingScript>, this element provides more information such as name of the script (text), start point of the crawl, if authentication is required or not, and such other details about the selenium script associated with the web application. Example:

```
<crawlingScripts>

              <count>1</count>

                 <list>

                  <SeleniumScript>

                   <id>2500</id>

                                <name><![CDATA[name of the
Script]]></name>

               <data>                .....

             <requiresAuthentication>

                true

              </requiresAuthentication>

             <startingUrl>URL</startingUrl>

              <startingUrlRegex>

                 true

              </startingUrlRegex>

          </SeleniumScript>

       </list> </crawlingScripts>
```

Elements Assigned by the Service	
id	(integer) The web application ID.
owner	(text) The user login ID of the web application owner.
isScheduled	(boolean) Is a scan scheduled for the web application? (true or false).

createdBy	(text) The user who created the web application.
createdDate	(date) The date when the web application was created in WAS, in UTC date/time format.
updatedBy	(text) The user who last updated the web application.
updatedDate	(date) The date when the web application was last updated in WAS, in UTC date/time format.
lastScan	(text) The scan ID of the last scan run on the web application.
lastScan.status	(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED

Authentication

Authentication Count

/qps/rest/3.0/count/was/webappauthrecord

[GET] [POST]

Returns the total number of authentication records in the user's scope. Input elements are optional and are used to filter the number of authentication records included in the count.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access" and Asset Management Permission "Read Asset". The output includes authentication records in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements

Click here for available operators

Parameter	Description
id	(integer) Authentication record ID.
name	(text) Authentication record name.
tags	(integer) Tag associated with the authentication record.
tags.name	(text) Tag name assigned to the authentication record.

tags.id	(integer) Tag ID assigned to the authentication record.
createdDate	(date) The date when the authentication record was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the authentication record was updated in WAS, in UTC date/time format.
lastScan.date	(date) The date when the web application (associated with the authentication record) was last scanned, in UTC date/time format.
lastScan.authStatus	(keyword) Authentication status reported by the last web application scan: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL
isUsed	(boolean) Indicates whether used by a web application or scan.
contents	(Keyword: FORM_STANDARD, FORM_CUSTOM, FORM_SELENIUM, SERVER_BASIC, SERVER_DIGEST, SERVER_NTLM, CERTIFICATE, OAUTH2_AUTH_CODE, OAUTH2_IMPLICIT, OAUTH2_PASSWORD, and OAUTH2_CLIENT_CREDS)

Sample - Get count of authentication records in user's account

Return the number (count) of all authentication records in the user’s scope.

API request

```
curl -u "USERNAME:PASSWORD"
https://qualysapi.qualys.com/qps/rest/3.0/count/was/webappauthrecord/"
```

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/3.
0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
</ServiceResponse>
```

Sample - Get count of authentication records with a criteria

Return the number (count) authentication records that have a name that contains the term “server”.

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">server</Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/webappauthrecord.xsd

Search Authentication Record

/qps/rest/3.0/search/was/webappauthrecord

[POST]

Returns a list of authentication records which are in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes authentication records in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements

The special field=attributes attribute for the Criteria element is used to search custom attributes (see sample below).

Click here for available operators

Parameter	Description
id	(integer) Authentication record ID.
name	(text) Authentication record name.
tags	(integer) Tag associated with the authentication record.
tags.name	(text) Tag name assigned to the authentication record.
tags.id	(integer) Tag ID assigned to the authentication record.

createdDate	(date) The date when the authentication record was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the authentication record was updated in WAS, in UTC date/time format.
lastScan.date	(date) The date when the web application (associated with the authentication record) was last scanned, in UTC date/time format.
lastScan.authStatus	(keyword) Authentication status reported by the last web application scan: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL
isUsed	(boolean) Indicates whether used by a web application or scan.
contents	(Keyword: FORM_STANDARD, FORM_CUSTOM, FORM_SELENIUM, SERVER_BASIC, SERVER_DIGEST, SERVER_NTLM, CERTIFICATE, OAUTH2_AUTH_CODE, OAUTH2_IMPLICIT, OAUTH2_PASSWORD, and OAUTH2_CLIENT_CREDS)

Samples

Sample - Search authentication records (no criteria)

Sample - Search for a particular authentication record

Sample - Search OAuth2 records with Implicit grant type

Sample - Search authentication records (no criteria)

Let us view a list of all authentication records in the user’s scope.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/webappauthrecord
/"
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.cm/qps/xsd/3.0
/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WebAppAuthRecord>
      <id>82605</id>
      <name><![CDATA[Form Only]]></name>
      <owner>
        <id>630926</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <tags>
        <count>3</count>
      </tags>
      <createdDate>2017-10-24T04:32:14Z</createdDate>
      <updatedDate>2017-10-24T07:45:05Z</updatedDate>
    </WebAppAuthRecord>
    <WebAppAuthRecord>
      <id>82606</id>
      ...
    </WebAppAuthRecord>
    <WebAppAuthRecord>
      <id>82607</id>
      ...
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

Sample - Search for a particular authentication record

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="EQUALS">82605</Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <WebAppAuthRecord>
            <id>82605</id>
            <name>
                <![CDATA[Sample auth]]>
            </name>
            <owner>
                <id>75913465</id>
                <username>username</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </owner>
            <tags>
                <count>0</count>
            </tags>
            <createdDate>2018-11-15T09:30:24Z</createdDate>
            <updatedDate>2018-11-15T09:30:24Z</updatedDate>
        </WebAppAuthRecord>
    </data>
</ServiceResponse>
```

Sample - Search OAuth2 records with Implicit grant type

Let us search OAuth2 records with Implicit grant type by passing OAUTH2_IMPLICIT keyword in the "contents" parameter.

API request

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

Request POST data

```
<ServiceRequest>
   <filters>
    <Criteria field="contents"
operator="IN">FORM_CUSTOM,SERVER_DIGEST,
    OAUTH2_IMPLICIT</Criteria>
   </filters>
</ServiceRequest>
<ServiceRequest>
   <filters>
    <Criteria field="contents" operator="EQUALS">OAUTH2_IMPLICIT
    </Criteria>
   </filters>
</ServiceRequest>
```

XML respons

XSD

<platform API server>/qps/xsd/3.0/was/webappauthrecord.xsd

Get Authentication Record Details

/qps/rest/3.0/get/was/webappauthrecord/<id>

[GET]

View details for an authentication record which is in the user's scope. Want to find a record ID to use as input? See Search authentication records.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes authentication records in the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies the authentication record.

Click here for available operators

Sample - View details for the authentication record

Let us view details for authentication record ID 74078.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webappauthrecord/74
078"
```

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/3.
0/was/webappauthrecord.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebAppAuthRecord>
      <id>74078</id>
      <name><![CDATA[My Authentication Record]]></name>
      <owner>
        <id>4354</id>
        <username>john_doe</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[does]]></lastName>
      </owner>
      <formRecord>
        <type>STANDARD</type>
        <sslOnly>true</sslOnly>
        <fields>
          <count>2</count>
          <list>
            <WebAppAuthFormRecordField>
              <id>826453</id>
              <name><![CDATA[name1]]></name>
              <value><![CDATA[value]]></value>
            </WebAppAuthFormRecordField>
            <WebAppAuthFormRecordField>
              <id>826452</id>
              <name><![CDATA[name2]]></name>
              <value><![CDATA[value]]></value>
            </WebAppAuthFormRecordField>
          </list>
        </fields>
      </formRecord>
      <tags>
        <count>1</count>
        <list>
          <Tag>
            <id>1418973</id>
            <name><![CDATA[Cert Tag]]></name>
          </Tag>
        </list>
      </tags>
      <comments>
        <count>0</count>
      </comments>
      <createdDate>2017-09-23T20:21:04Z</createdDate>
      <createdBy>
        <id>4354</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <updatedDate>2017-10-22T05:48:57Z</updatedDate>
      <updatedBy>
        <id>4354</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

Sample - Password is masked

Let us fetch authentication record details with the password fields masked when sub user has disabled "View Password in Authentication Record" and "View/download Selenium Script sensitive contents" permissions.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webappauthrecord/76
1533"
```

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/3.
0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebAppAuthRecord>
            <id>761533</id>
            <name><![CDATA[Selenium record]]></name>
            <owner>
                <id>75670165</id>
                <username>john_doe </username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </owner>
            <formRecord>
```

```
                <type>SELENIUM</type>
                <seleniumScript>
                    <name>
                        <![CDATA[seleniumScript]]>
                    </name>
                    <data>
                        <![CDATA[
                        <?xml version="1.0" encoding="UTF-8"?>
                        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
                        <html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
                            <head>
                                <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
                                <link rel="selenium.base"
href="https://10.113.195.231/" />
                                <title>AuthScript</title>
                            </head>
                            <body>
                                <table cellpadding="1" cellspacing="1"
border="1">
                                    <thead>
                                        <tr>
                                            <td rowspan="1"
colspan="3">AuthScript</td>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <tr>
                                            <td>open</td>
                                            <td>@@webappURL@@</td>
                                            <td></td>
                                        </tr>
                                        <tr>
                                            <td>click</td>
                                            <td>name=username</td>
                                            <td></td>
                                        </tr>
                                        <tr>
                                            <td>type</td>
                                            <td>name=username</td>
                                            <td>*****</td>
                                        </tr>
                                        <tr>
```

```
                                            <td>type</td>
                                            <td>name=password</td>
                                            <td>*****</td>
                                        </tr>
                                        <tr>
                                            <td>click</td>
                                            <td>name=Login</td>
                                            <td></td>
                                        </tr>
                                    </tbody>
                                </table>
                            </body></html>]]>
                        </data>
                        <regex>
                            <![CDATA[selenium]]>
                        </regex>
                    </seleniumScript>
                </formRecord>
                <serverRecord>
                    <fields>
                        <count>3</count>
                        <list>
                            <WebAppAuthServerRecordField>
                                <id>730020</id>
                                <type>BASIC</type>
                                <domain>
                                    <![CDATA[comp]]>
                                </domain>
                                <username>
                                    <![CDATA[abc]]>
                                </username>
                                <password>
                                    <![CDATA[*****]]>
                                </password>
                            </WebAppAuthServerRecordField>
                            <WebAppAuthServerRecordField>
                                <id>730021</id>
                                <type>NTLM</type>
                                <username>
                                    <![CDATA[abc3]]>
                                </username>
                                <password>
                                    <![CDATA[*****]]>
                                </password>
                            </WebAppAuthServerRecordField>
                            <WebAppAuthServerRecordField>
                                <id>730022</id>
                                <type>DIGEST</type>
                                <domain>
                                    <![CDATA[comp2]]>
                                </domain>
                                <username>
                                    <![CDATA[abc2]]>
                                </username>
                                <password>
                                    <![CDATA[*****]]>
                                </password>
                            </WebAppAuthServerRecordField>
                        </list>
                    </fields>
                </serverRecord>
               ...
                </updatedBy>
            </WebAppAuthRecord>
        </data>
    </ServiceResponse>
```

Sample - Password is visible

Let us fetch authentication record details with the password fields visible when sub user has disabled "View Password in Authentication Record" and "View/download Selenium Script sensitive contents" permissions.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webappauthrecord/76
1534"
```

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/3.
0/was/webappauthrecord.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebAppAuthRecord>
            <id>762380</id>
            <name>
                <![CDATA[Selenium with server authentication]]>
            </name>
            <owner>
                <id>75913465</id>
                <username>john_doe</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[doe]]>
                </lastName>
            </owner>
            <formRecord>
                <type>SELENIUM</type>
                <seleniumScript>
                    <name>
                        <![CDATA[seleniumScript]]>
                    </name>
                    <data>
                        <![CDATA[
                        <?xml version="1.0" encoding="UTF-8"?>
                        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
                        <html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
                            <head>
                                <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
                                <link rel="selenium.base"
href="https://10.113.195.231/" />
                                <title>AuthScript</title>
                            </head>
                            <body>
                                <table cellpadding="1" cellspacing="1"
border="1">
                                    <thead>
                                        <tr>
                                            <td rowspan="1"
colspan="3">AuthScript</td>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <tr>
                                            <td>open</td>
                                            <td>@@webappURL@@</td>
                                            <td></td>
                                        </tr>
                                        <tr>
                                            <td>click</td>
                                            <td>name=username</td>
                                            <td></td>
                                        </tr>
                                        <tr>
                                            <td>type</td>
                                            <td>name=username</td>
                                            <td>theuser</td>
                                        </tr>
                                        <tr>
                                            <td>type</td>
                                            <td>name=password</td>
                                            <td>thepass</td>
                                        </tr>
                                        <tr>
                                            <td>click</td>
                                            <td>name=Login</td>
                                            <td></td>
                                        </tr>
                                    </tbody>
                                </table>
                            </body></html>]]>
                        </data>
                        <regex>
                            <![CDATA[selenium]]>
                        </regex>
                    </seleniumScript>
                </formRecord>
                <serverRecord>
                    <fields>
                        <count>3</count>
                        <list>
                            <WebAppAuthServerRecordField>
                                <id>731073</id>
                                <type>NTLM</type>
                                <username>
                                    <![CDATA[abc3]]>
                                </username>
                                <password>
                                    <![CDATA[1234]]>
                                </password>
                            </WebAppAuthServerRecordField>
                            <WebAppAuthServerRecordField>
                                <id>731074</id>
                                <type>BASIC</type>
                                <domain>
                                    <![CDATA[comp]]>
                                </domain>
                                <username>
                                    <![CDATA[abc]]>
                                </username>
                                <password>
                                    <![CDATA[1234]]>
                                </password>
                            </WebAppAuthServerRecordField>
                            <WebAppAuthServerRecordField>
                                <id>731075</id>
                                <type>DIGEST</type>
                                <domain>
                                    <![CDATA[comp2]]>
                                </domain>
                                <username>
                                    <![CDATA[abc2]]>
                                </username>
                                <password>
                                    <![CDATA[1234]]>
                                </password>
                            </WebAppAuthServerRecordField>
                        </list>
                    </fields>
                </serverRecord>
                ....
                </updatedBy>
            </WebAppAuthRecord>
        </data>
    </ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/webappauthrecord.xsd

Create Authentication Record

/qps/rest/3.0/create/was/webappauthrecord

[POST]

Creates a new authentication record.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and Asset Management Permission “Create Authentication Record”. The output includes authentication records in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements.

Click here for available operators

Parameter	Description
name	(text) Authentication record name.
WebAppAuthRecord	(text) Details associated with the web application authentication record. Use these parameters to create OAuth2 authentication record: WebAppAuthRecord.oauth2Record.grantType - (text) (Required if authentication type is OAuth2) Valid values are: 1) NONE, AUTH_CODE, IMPLICIT, PASSWORD, and CLIENT_CREDS. NONE means no grant type is selected. These are fields we support for each grant type:

1) AUTH_CODE - We support these fields for Authorization Code: 1) seleniumScript, 2) redirectUrl, 3) accessTokenUrl, 4) clientId (optional), 5) clientSecret (optional), 6) scope, (optional) and 7) accessTokenExpiredMsgPattern (optional)

Note: Selenium script is mandatory for Authorization Code. We support parametrized username and password in the selenium script. See “Create a Selenium script to parameterize username and password” in the WAS API guide.

2) IMPLICIT - We support these fields for Implicit: 1) seleniumScript, and 2) redirectUrl

Note: Selenium script is mandatory for Implicit. We support parametrized username and password in the selenium script. See “Create a Selenium script to parameterize username and password” in the WAS API guide.

3) PASSWORD - We support these fields for Resource Owner Password Credentials: 1) accessTokenUrl, 2) username, 3) password, 4) clientId (optional), 5) clientSecret (optional), 6) scope (optional), and 7) accessTokenExpiredMsgPattern (optional)

4) CLIENT_CREDS - We support these fields for Client Credentials: 1) accessTokenUrl, 2) clientId (optional), 3) clientSecret (optional), and 4) scope, (optional)

Note:

When creating an authentication record, you can specify either a Form record (used for web application authentication) or an OAuth2 record (used for the Swagger/Open API file authentication) in the request. While updating an authentication record,

	- Send the Form record with type as NONE if you want to set an OAuth2 record instead of a form record. - Send OAuth2 with grant type as NONE if you want to set a Form record instead of an OAuth2 record.
tags	(text) Tag associated with the authentication record.
comments	(text) User-defined comments.

Samples

Sample - Create a standard authentication record

Sample - Create a custom authentication record

Sample - Create a Selenium script

Sample - Create a Selenium script to parameterize username and password

Sample - Create server authentication

Sample: Create an OAuth2 authentication record with grant type as Client Credentials

Sample: Create an OAuth2 authentication record with Selenium script

Sample - Create a standard authentication record

Let us create a new web application called “My Web Application” that has the starting URL “http://mywebapp.com”. The default web application settings are assigned automatically.

API request

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

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

Request POST data

```
<ServiceRequest>
  <data>
  <WebAppAuthRecord>
    <name><![CDATA[STANDARD auth]]></name>
    <formRecord>
      <type>STANDARD</type>
      <sslOnly>true</sslOnly>
      <fields>
        <set>
          <WebAppAuthFormRecordField>
            <name>username</name>
            <value>john</value>
          </WebAppAuthFormRecordField>
          <WebAppAuthFormRecordField>
            <name>password</name>
            <value>secret</value>
          </WebAppAuthFormRecordField>
        </set>
      </fields>
    </formRecord>
    <tags>
      <set>
        <Tag>
          <id>152743</id>
        </Tag>
      </set>
    </tags>
    <comments>
      <set>
        <Comment><contents><![CDATA[some
comments]]></contents></Comment>
      </set>
    </comments>
  </WebAppAuthRecord>
  </data>
</ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebAppAuthRecord>
      <id>80149</id>
      <name><![CDATA[STANDARD auth]]></name>
      <owner>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <formRecord>
        <type>STANDARD</type>
        <sslOnly>true</sslOnly>
        <fields>
          <count>2</count>
          <list>
            <WebAppAuthFormRecordField>
              <id>835050</id>
              <name><![CDATA[username]]</name>
            <value><![CDATA[john]]</value>
            </WebAppAuthFormRecordField>
            <WebAppAuthFormRecordField>
              <id>835051</id>
              <name><![CDATA[username]]></name>
              <value><![CDATA[jim]]></value>
            </WebAppAuthFormRecordField>
          </list>
        </fields>
      </formRecord>
      <tags>
        <count>1</count>
        <list>
          <Tag>
            <id>152743</id>
            <name><![CDATA[Asset Groups]]></name>
          </Tag>
        </list>
      </tags>
      <comments>
        <count>1</count>
        <list>
          <Comment>
            <contents><![CDATA[some comments]]></contents>
            <createdDate>2017-10-18T18:18:01Z</createdDate>
          </Comment>
        </list>
      </comments>
      <createdDate>2017-10-18T18:18:01Z</createdDate>
      <createdBy>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <updatedDate>2017-10-18T18:18:01Z</updatedDate>
      <updatedBy>
        <id>45941</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

Sample - Create a custom authentication record

API request

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

Request POST data

```
<ServiceRequest>
  <data>
  <WebAppAuthRecord>
    <name><![CDATA[CUSTOM auth]]></name>
      <formRecord>
        <type>CUSTOM</type>
        <sslOnly>true</sslOnly>
        <fields>
```

```
          <set>
            <WebAppAuthFormRecordField>
              <name>some username</name>
              <value>Login</value>
              <secured>false</secured>
            </WebAppAuthFormRecordField>
            <WebAppAuthFormRecordField>
              <name>some password with true</name>
              <value>real password</value>
              <secured>true</secured>
            </WebAppAuthFormRecordField>
            <WebAppAuthFormRecordField>
              <name>not password with false</name>
              <secured>false</secured>
              <value>fake password</value>
            </WebAppAuthFormRecordField>
           </set>
         </fields>
       </formRecord>
       <comments>
         <set>
           <Comment><contents><![CDATA[some
comments]]></contents></Comment>
         </set>
       </comments>
  </WebAppAuthRecord>
  </data>
</ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebAppAuthRecord>
            <id>685133</id>
            <name><![CDATA[CUSTOM auth]]></name>
            <owner>
                <id>75913465</id>
                <username>username</username>
                <firstName> <![CDATA[John]]></firstName>
                <lastName><![CDATA[Smith]]></lastName>
            </owner>
            <formRecord>
                <type>CUSTOM</type>
                <sslOnly>true</sslOnly>
                <fields>
                    <count>3</count>
                    <list>
                        <WebAppAuthFormRecordField>
                            <id>692981</id>
                            <name><![CDATA[not password with 
false]]></name>
                            <secured>false</secured>
                            <value><![CDATA[fake password]]></value>
                        </WebAppAuthFormRecordField>
                        <WebAppAuthFormRecordField>
                            <id>692982</id>
                            <name><![CDATA[some password with 
true]]></name>
                            <secured>true</secured>
                            <value><![CDATA[*****]]></value>
                        </WebAppAuthFormRecordField>
                        <WebAppAuthFormRecordField>
                            <id>692983</id>
                            <name><![CDATA[some username]]></name>
                            <secured>false</secured>
                            <value><![CDATA[Login]]></value>
                        </WebAppAuthFormRecordField>
                    </list>
                </fields>
            </formRecord>
            <tags>
                <count>0</count>
            </tags>
            <comments>
                <count>1</count>
                <list>
                    <Comment>
                        <contents>
                            <![CDATA[some comments]]>
                        </contents>
                        <createdDate>2018-11-
21T09:25:00Z</createdDate>
                    </Comment>
                </list>
            </comments>
            <createdDate>2018-11-21T09:25:00Z</createdDate>
            <createdBy>
                <id>75913465</id>
                <username>username</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </createdBy>
            <updatedDate>2018-11-21T09:25:00Z</updatedDate>
            <updatedBy>
                <id>75913465</id>
                <username>username</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </updatedBy>
        </WebAppAuthRecord>
    </data>
</ServiceResponse>
```

Sample - Create a Selenium script

API request

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <data>
    <WebAppAuthRecord>
      <name><![CDATA[From API - Selenium]]></name>
      <formRecord>
```

```
        <type>SELENIUM</type>
        <seleniumScript>
          <name><![CDATA[seleniumScriptOK]]></name>
          <data><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="https://community.qualys.com/" />
<title>seleniumScriptOK</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">seleniumScriptOK</td></tr>
</thead><tbody>
<tr>
    <td>open</td>
    <td>https://community.qualys.com/index.jspa</td>
    <td></td>
</tr>
<tr>
    <td>clickAndWait</td>
    <td>css=#qc-homepage-cafe > span.qc-homepage-header-item-
title</td>
    <td></td>
</tr>
<tr>
    <td>clickAndWait</td>
    <td>link=Introduction to Qualys Mapping</td>
    <td></td>
</tr>
</tbody></table>
</body>
</html>]]></data>
          <regex><![CDATA[selenium]]></regex>
        </seleniumScript>
      </formRecord>
    </WebAppAuthRecord>
  </data>
</ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebAppAuthRecord>
            <id>307757</id>
            <name>
                <![CDATA[From API - Selenium]]>
            </name>
            <owner>
                <id>4354</id>
                <username>user_alice</username>
                <firstName>
                    <![CDATA[Alice]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </owner>
            <formRecord>
                <type>SELENIUM</type>
                <seleniumScript>
                    <name>
                        <![CDATA[seleniumScriptOK]]>
                    </name>
                    <data>
                        <![CDATA[
                        <?xml version="1.0" encoding="UTF-8"?>
                        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
                        <html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
                            <head profile="http://selenium-
ide.openqa.org/profiles/test-case">
                                <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
                                <link rel="selenium.base"
href="https://community.qualys.com/" />
                                <title>seleniumScriptOK</title>
                            </head>
                            <body>
```

```
                                <table cellpadding="1" cellspacing="1"
border="1">
                                    <thead>
                                        <tr>
                                            <td rowspan="1"
colspan="3">seleniumScriptOK</td>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <tr>
                                            <td>open</td>
                                            <td>https://community.qual
ys.com/index.jspa</td>
                                            <td></td>
                                        </tr>
                                        <tr>
                                            <td>clickAndWait</td>
                                            <td>css=#qc-homepage-cafe
> span.qc-homepage-header-item-title</td>
                                            <td></td>
                                        </tr>
                                        <tr>
                                            <td>clickAndWait</td>
                                            <td>link=Introduction to
Qualys Mapping</td>
                                            <td></td>
                                        </tr>
                                    </tbody>
                                </table>
                            </body></html>]]>
                        </data>
                        <regex>
                            <![CDATA[selenium]]>
                        </regex>
                    </seleniumScript>
                </formRecord>
                <tags>
                    <count>0</count>
                </tags>
                <comments>
                    <count>0</count>
                </comments>
                <createdDate>2017-05-06T16:23:43Z</createdDate>
                <createdBy>
                    <id>4354</id>
                    <username>user_alex</username>
                    <firstName>
                        <![CDATA[Alice]]>
                    </firstName>
                    <lastName>
                        <![CDATA[Smith]]>
                    </lastName>
                </createdBy>
                <updatedDate>2017-05-06T16:23:43Z</updatedDate>
                <updatedBy>
                    <id>4354</id>
                    <username>user_alex</username>
                    <firstName>
                        <![CDATA[Alice]]>
                    </firstName>
                    <lastName>
                        <![CDATA[Smith]]>
                    </lastName>
                </updatedBy>
            </WebAppAuthRecord>
        </data>
    </ServiceResponse>
```

Sample - Create a Selenium script to parameterize username and password

When using selenium script for authentication, you have the option to parameterize the username and password. Specify the username and password in the authentication record and then during the scan, we will replace @@authusername@@ and @@authpassword@@ with this username and password. Add these 2 parameters: @@authusername@@ for username and @@authpassword@@ for password inside the Selenium script.
The parameter names are case insensitive.

The advantage of using the parameters in the script is that you can change the login credentials without modifying your selenium script.

To use the parameters inside the selenium script, you need to set “seleniumCreds” to “true” in the authentication record. If you set the parameter to “false”, then adding the placeholders in the script will return an error.

Let us create an authentication record of type Selenium script and add @@authusername@@ and @@authpassword@@ inside the selenium script and set the parameter “seleniumCreds” to ″true″.

API request

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
   <data>
      <WebAppAuthRecord>
         <name><![CDATA[From API - Selenium]]></name>
         <formRecord>
            <type>SELENIUM</type>
            <seleniumScript>
               <name><![CDATA[seleniumScriptOK]]></name>
               <data><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://10.10.31.25/" />
<title>seleauth</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">Untitled Test Case</td></tr>
</thead>
<tbody>
<tr><td>open</td><td>http://10.10.31.25/login_2/index.php</td><td></td
>
</tr>
<tr><td>type</td><td>name=username</td><td>@@authusername@@</td>
</tr>
<tr><td>type</td><td>name=password</td><td>@@authpassword@@</td>
</tr>
<tr><td>click</td><td>css=input[type="submit"]</td><td></td>
</tr>
</tbody></table>
</body>
</html>]]></data>
               <regex><![CDATA[selenium]]></regex>
            </seleniumScript>
            <seleniumCreds>true</seleniumCreds>
            <fields>
               <set>
                  <WebAppAuthFormRecordField>
                     <name>username</name>
                     <value>spp2</value>
                  </WebAppAuthFormRecordField>
                  <WebAppAuthFormRecordField>
                     <name>password</name>
                     <value>secret</value>
                  </WebAppAuthFormRecordField>
               </set>
            </fields>
         </formRecord>
      </WebAppAuthRecord>
   </data>
</ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebAppAuthRecord>
            <id>804942</id>
            <name>
                <![CDATA[From API - Selenium]]>
            </name>
            <owner>
                <id>5759808</id>
                <username>joe_user</username>
                <firstName>
                    <![CDATA[Sunny]]>
                </firstName>
                <lastName>
                    <![CDATA[Mirani]]>
                </lastName>
            </owner>
            <formRecord>
                <type>SELENIUM</type>
                <authVault>false</authVault>
                <seleniumCreds>true</seleniumCreds>
                <seleniumScript>
                    <name>
                        <![CDATA[
                        seleniumScriptOK
                    ]]>
                    </name>
                    <data>
                        <![CDATA[
                        <?xml version="1.0" encoding="UTF-8"?>
                        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
                        <html xmlns="http://www.w3.org/1999/xhtml" 
xml:lang="en" lang="en">
                            <head>
                                <meta http-equiv="Content-Type" 
content="text/html; charset=UTF-8" />
                                <link rel="selenium.base" 
href="http://10.10.31.25/" />
                                <title>seleauth</title>
                            </head>
                            <body>
                                <table cellpadding="1" cellspacing="1" 
border="1">
                                    <thead>
                                        <tr>
                                            <td rowspan="1" 
colspan="3">Untitled Test Case</td>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <tr>
                                            <td>open</td>
                                            <td>http://10.10.31.25/log
in_2/index.php</td>
                                            <td></td>
                                        </tr>
                                        <tr>
```

```
                                            <td>type</td>
                                            <td>name=username</td>
                                            <td>@@authusername@@</td>
                                        </tr>
                                        <tr>
                                            <td>type</td>
                                            <td>name=password</td>
                                            <td>@@authpassword@@</td>
                                        </tr>
                                        <tr>
                                            <td>click</td>
                                            <td>css=input[type="submit
"]</td>
                                            <td></td>
                                        </tr>
                                    </tbody>
                                </table>
                            </body></html>]]>
                        </data>
                        <regex>
                            <![CDATA[
                            selenium
                        ]]>
                        </regex>
                    </seleniumScript>
                    <fields>
                        <count>2</count>
                        <list>
                            <WebAppAuthFormRecordField>
                                <id>860000</id>
                                <name>
                                    <![CDATA[PASSWORD]]>
                                </name>
                                <secured>true</secured>
                                <value>
                                    <![CDATA[*****]]>
                                </value>
                            </WebAppAuthFormRecordField>
                            <WebAppAuthFormRecordField>
                                <id>860001</id>
                                <name>
                                    <![CDATA[USERNAME]]>
                                </name>
                                <secured>false</secured>
                                <value>
```

```
                                    <![CDATA[spp215]]>
                                </value>
                            </WebAppAuthFormRecordField>
                        </list>
                    </fields>
                </formRecord>
                <tags>
                    <count>0</count>
                </tags>
                <comments>
                    <count>0</count>
                </comments>
                <createdDate>2021-06-01T04:18:38Z</createdDate>
                <createdBy>
                    <id>5759808</id>
                    <username>joe_user</username>
                    <firstName>
                        <![CDATA[joe]]>
                    </firstName>
                    <lastName>
                        <![CDATA[user]]>
                    </lastName>
                </createdBy>
                <updatedDate>2021-06-014T04:18:38Z</updatedDate>
                <updatedBy>
                    <id>5759808</id>
                    <username>joe_user</username>
                    <firstName>
                        <![CDATA[joe]]>
                    </firstName>
                    <lastName>
                        <![CDATA[user]]>
                    </lastName>
                </updatedBy>
            </WebAppAuthRecord>
        </data>
    </ServiceResponse>
```

Sample - Create server authentication

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
```

```
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/webappauthrecord
/" < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <data>
  <WebAppAuthRecord>
    <name><![CDATA[server auth]]></name>
      <serverRecord>
        <sslOnly>true</sslOnly>
        <certificate>
          <name><![CDATA[My Certificate]]></name>
<contents><![CDATA[-----BEGIN CERTIFICATE-----
MIIC4jCCAkugAwIBAgIJAPU+Kw6GX2aMMA0GCSqGSIb3DQEBBQUAMIGJMQswCQYD
VQQGEwJGUjEPMA0GA1UECAwGRnJhbmNlMREwDwYDVQQHDAhUb3Vsb3VzZTEPMA0G
A1UECgwGUXVhbHlzMRUwEwYDVQQLDAxRdWFseXMgVGVjaC4xDTALBgNVBAMMBE5p
Y28xHzAdBgkqhkiG9w0BCQEWEG5iaXplQHF1YWx5cy5jb20wHhcNMTExMDA1MjIx
...
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC4SiB/HaNxQtwQUtot867MxTP1PqAQh7VyHIdBs037eafpd8B6
apHhih0Jw0zr2RzcWniUUhhpvwL4apG470/RzkIKSNu4h9akHqA5b0Pe0ZasrE7B
MxUZWNf9dfrY+JXQmdaPce0i4w4zZR+PabXDy5Mg9ONEUKS3AONCHk7acwIDAQAB
AoGAMHwAFLFdgLzQXNMPZ6uGv4TaaJkzT2YEzKLIyvY7e//Dt160GwDSpH3Lqffh
...
-----END RSA PRIVATE KEY-----]]></contents>
          <passphrase>My Certificate</passphrase>
        </certificate>
      </serverRecord>
    <comments>
      <set>
        <Comment><contents><![CDATA[some
comments]]></contents></Comment>
      </set>
    </comments>
  </WebAppAuthRecord>
  </data>
</ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebAppAuthRecord>
            <id>685134</id>
            <name>
                <![CDATA[server auth]]>
            </name>
            <owner>
                <id>75913465</id>
                <username>user_john</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </owner>
            <serverRecord>
                <sslOnly>true</sslOnly>
                <certificate>
                    <name>
                        <![CDATA[My Certificate]]>
                    </name>
                    <contents>
                        <![CDATA[-----BEGIN CERTIFICATE-----
MIIC4jCCAkugAwIBAgIJAPU+Kw6GX2aMMA0GCSqGSIb3DQEBBQUAMIGJMQswCQYD
VQQGEwJGUjEPMA0GA1UECAwGRnJhbmNlMREwDwYDVQQHDAhUb3Vsb3VzZTEPMA0G
A1UECgwGUXVhbHlzMRUwEwYDVQQLDAxRdWFseXMgVGVjaC4xDTALBgNVBAMMBE5p
Y28xHzAdBgkqhkiG9w0BCQEWEG5iaXplQHF1YWx5cy5jb20wHhcNMTExMDA1MjIx
OTQ5WhcNMTIxMDA0MjIxOTQ5WjCBiTELMAkGA1UEBhMCRlIxDzANBgNVBAgMBkZy
YW5jZTERMA8GA1UEBwwIVG91bG91c2UxDzANBgNVBAoMBlF1YWx5czEVMBMGA1UE
CwwMUXVhbHlzIFRlY2guMQ0wCwYDVQQDDAROaWNvMR8wHQYJKoZIhvcNAQkBFhBu
Yml6ZUBxdWFseXMuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC4SiB/
HaNxQtwQUtot867MxTP1PqAQh7VyHIdBs037eafpd8B6apHhih0Jw0zr2RzcWniU
UhhpvwL4apG470/RzkIKSNu4h9akHqA5b0Pe0ZasrE7BMxUZWNf9dfrY+JXQmdaP
ce0i4w4zZR+PabXDy5Mg9ONEUKS3AONCHk7acwIDAQABo1AwTjAdBgNVHQ4EFgQU
2MfOD/CeGujBKQGGaXTIoPPUPUowHwYDVR0jBBgwFoAU2MfOD/CeGujBKQGGaXTI
oPPUPUowDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOBgQCySmA+hoaTvmlV
RptWqID24SH3r4FQPR9wan/RcXbpTl5hwboauUspqwSmx/jKKtsLy9VdLJLnf3NT
p/9U5TscI5d3Xw5EHKVcNnGFZwb5uc18jXhvEpUi+WWlGhIP6Nin3gdAcyxXYTcB
ILvK0dar1//cJZRn7+tRFFcw6keITQ==
-----END CERTIFICATE-----
]]>
                    </contents>
                </certificate>
                <fields>
                    <count>0</count>
                </fields>
            </serverRecord>
            <tags>
                <count>0</count>
            </tags>
            <comments>
                <count>1</count>
                <list>
                    <Comment>
                        <contents>
                            <![CDATA[some comments]]>
                        </contents>
                        <createdDate>2018-11-
21T09:41:59Z</createdDate>
                    </Comment>
                </list>
            </comments>
            <createdDate>2018-11-21T09:41:59Z</createdDate>
            <createdBy>
                <id>75913465</id>
                <username>user_john</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </createdBy>
            <updatedDate>2018-11-21T09:41:59Z</updatedDate>
            <updatedBy>
                <id>75913465</id>
                <username>username</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </updatedBy>
```

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

Sample - Create an OAuth2 authentication record with grant type as Client Credentials

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <WebAppAuthRecord>
            <name><![CDATA[GrantType-Client-credentials-SPP]]></name>
            <oauth2Record>
                <grantType>CLIENT_CREDS</grantType>
                <accessTokenUrl>http://www.authTokenUrl.com
                </accessTokenUrl>
                <clientId>clientIdVal</clientId>
                <clientSecret>clientSecretVal</clientSecret>
                <scope>scope</scope>
            </oauth2Record>
        </WebAppAuthRecord>
    </data>
</ServiceRequest>
```

Sample - Create an OAuth2 authentication record with Selenium script

Let us create an OAuth2 authentication record with grant type Implicit that requires selenium script.

API request

```
curl -n -u "USERNAME:PASSWORD" -H "content-type: text/xml"-X "POST" --
data-binary @-
```

```
"https://qualysapi.qualys.com/rest/3.0/create/was/webappauthrecord" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <data>
        <WebAppAuthRecord>
            <name>
                <![CDATA[OAuth2 and Server Auth Record]]>
            </name>
            <serverRecord>
                <sslOnly>true</sslOnly>
                <fields>
                    <set>
                        <WebAppAuthServerRecordField>
                            <type>DIGEST</type>
                            <domain>realm</domain>
                            <username>
                                <![CDATA[username]]>
                            </username>
                            <password>password</password>
                        </WebAppAuthServerRecordField>
                    </set>
                </fields>
            </serverRecord>
            <oauth2Record>
                <grantType>IMPLICIT</grantType>
                <redirectUrl>http://www.redirectUrl.com</redirectUrl>
                <seleniumScript>
                    <name>
                        <![CDATA[seleniumScriptOK]]>
                    </name>
                    <data>
                        <![CDATA[
                        <?xml version="1.0" encoding="UTF-8"?>
                        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
                        <html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
                            <head>
                                <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
```

```
                                <link rel="selenium.base"
href="http://10.10.31.25/" />
                                <title>seleauth</title>
                            </head>
                            <body>
                                <table cellpadding="1" cellspacing="1"
border="1">
                                    <thead>
                                        <tr>
                                            <td rowspan="1"
colspan="3">Untitled Test Case</td>
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <tr>
                                            <td>open</td>
                                            <td>http://10.10.31.25/log
in_2/index.php</td>
                                            <td></td>
                                        </tr>
                                        <tr>
                                            <td>type</td>
                                            <td>name=username</td>
                                            <td>@@authusername@@</td>
                                        </tr>
                                        <tr>
                                            <td>type</td>
                                            <td>name=password</td>
                                            <td>@@authpassword@@</td>
                                        </tr>
                                        <tr>
                                            <td>click</td>
                                            <td>css=input[type="submit
"]</td>
                                            <td></td>
                                        </tr>
                                    </tbody>
                                </table>
                            </body></html>]]>
                        </data>
                        <regex>
                            <![CDATA[selenium]]>
                        </regex>
                    </seleniumScript>
                    <seleniumCreds>true</seleniumCreds>
                    <username>uname</username>
                    <password>pwd</password>
                </oauth2Record>
            </WebAppAuthRecord>
        </data>
```

XSD

<platform API server>/qps/xsd/3.0/was/webappauthrecord.xsd

Update Authentication Record

/qps/rest/3.0/update/was/webappauthrecord/<id>

[POST]

Update an authentication record which is in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes authentication records in the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies an authentication record.

Click here for available operators

Samples

Sample - Update authentication record settings

Sample: Update a Form authentication record to OAuth2 record

Sample: Update a Form authentication record to OAuth2 record with selenium script

Sample - Update authentication record settings

Let us update the settings for authentication record ID 82605.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/webappauthrecord
/82605" < file.xml
Note: "file.xml" contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <data>
    <WebAppAuthRecord>
      <name><![CDATA[Form and Server Auth]]></name>
      <serverRecord>
        <sslOnly>true</sslOnly>
        <fields>
          <set>
            <WebAppAuthServerRecordField>
              <type>DIGEST</type>
              <domain>realm</domain>
              <username><![CDATA[username]]></username>
              <password>password</password>
            </WebAppAuthServerRecordField>
          </set>
        </fields>
      </serverRecord>
      <formRecord>
        <type>STANDARD</type>
        <sslOnly>true</sslOnly>
        <fields>
          <set>
            <WebAppAuthFormRecordField>
              <name>username</name>
              <value>Login</value>
            </WebAppAuthFormRecordField>
          </set>
        </fields>
      </formRecord>
    </WebAppAuthRecord>
  </data>
</ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebAppAuthRecord>
      <id>82605</id>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

Sample: Update a Form authentication record to OAuth2 record

Let us update a form authentication record to set OAuth2 record with Client Credentials grant type. If you want to set an OAuth2 record instead of a form record, then set the form record with type as NONE.

API request

```
curl -n -u "USERNAME:PASSWORD" -H "content-type: text/xml"-X "POST" --
data-binary @-
"https://qualysapi.qualys.com/rest/3.0/update/was/webappauthrecord/826
09" < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <data>
        <WebAppAuthRecord>
            <name><![CDATA[My Oauth Record]]></name>
            <serverRecord>
                <sslOnly>true</sslOnly>
                <fields>
                    <set>
                        <WebAppAuthServerRecordField>
                            <type>DIGEST</type>
                            <domain>realm</domain>
                            <username><![CDATA[username]]></username>
                            <password>password</password>
                        </WebAppAuthServerRecordField>
                    </set>
                </fields>
            </serverRecord>
            <formRecord>
                <type>NONE</type>
            </formRecord>
            <oauth2Record>
                <grantType>CLIENT_CREDS</grantType>
                <accessTokenUrl>http://www.authTokenUrl.com
                </accessTokenUrl>
                <clientId>clientIdVal</clientId>
                <clientSecret>clientSecretVal</clientSecret>
                <scope>scope</scope>
            </oauth2Record>
        </WebAppAuthRecord>
    </data>
</ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebAppAuthRecord>
      <id>82609</id>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

Sample: Update a Form authentication record to OAuth2 record with selenium script

Let us update a form authentication record to set OAuth2 record with grant type Implicit that requires selenium script. If you want to set an OAuth2 record instead of a form record, then set the form record with type as NONE.

API request

```
curl -n -u "USERNAME:PASSWORD" -H "content-type: text/xml"-X "POST" --
data-binary @-
"https://qualysapi.qualys.com/rest/3.0/update/was/webappauthrecord/826
22" < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <data>
        <WebAppAuthRecord>
            <name>
                <![CDATA[OAuth2 and Server Auth Record]]>
            </name>
            <serverRecord>
                <sslOnly>true</sslOnly>
                <fields>
                    <set>
                        <WebAppAuthServerRecordField>
                            <type>DIGEST</type>
                            <domain>realm</domain>
                            <username>
                                <![CDATA[username]]>
                            </username>
                            <password>password</password>
                        </WebAppAuthServerRecordField>
                    </set>
                </fields>
            </serverRecord>
            <oauth2Record>
                <grantType>IMPLICIT</grantType>
                <redirectUrl>http://www.redirectUrl.com</redirectUrl>
                <seleniumScript>
                    <name>
                        <![CDATA[seleniumScriptOK]]>
                    </name>
                    <data>
                        <![CDATA[
                        <?xml version="1.0" encoding="UTF-8"?>
                        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
                         strict.dtd">
                        <html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
                            <head>
                                <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
                                <link rel="selenium.base"
href="http://10.10.31.25/" />
                                <title>seleauth</title>
                            </head>
                            <body>
                                <table cellpadding="1" cellspacing="1"
border="1">
                                    <thead>
                                        <tr>
                                            <td rowspan="1"
colspan="3">Untitled Test Case</td>
```

```
                                        </tr>
                                    </thead>
                                    <tbody>
                                        <tr>
                                            <td>open</td>
                                            <td>http://10.10.31.25/log
in_2/index.php</td>
                                            <td></td>
                                        </tr>
                                        <tr>
                                            <td>type</td>
                                            <td>name=username</td>
                                            <td>@@authusername@@</td>
                                        </tr>
                                        <tr>
                                            <td>type</td>
                                            <td>name=password</td>
                                            <td>@@authpassword@@</td>
                                        </tr>
                                        <tr>
                                            <td>click</td>
                                            <td>css=input[type="submit
"]</td>
                                            <td></td>
                                        </tr>
                                    </tbody>
                                </table>
                            </body></html>]]>
                        </data>
                        <regex>
                            <![CDATA[selenium]]>
                        </regex>
                    </seleniumScript>
                    <seleniumCreds>true</seleniumCreds>
                    <username>uname</username>
                    <password>pwd</password>
                </oauth2Record>
            </WebAppAuthRecord>
        </data>
    </ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebAppAuthRecord>
      <id>82622</id>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/webappauthrecord.xsd

Delete Authentication Record

/qps/rest/3.0/delete/was/webappauthrecord/<id>

/qps/rest/3.0/delete/was/webappauthrecord/<filters>

[POST]

Delete an authentication record which is in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The authentication record to be deleted must be within the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements

[Click here for available operators]

Parameter	Description
id	(integer) Authentication record ID.
name	(text) Authentication record name.
tags	(integer) Tag associated with the authentication record.
tags.name	(text) Tag name assigned to the authentication record.
tags.id	(integer) Tag ID assigned to the authentication record.
createdDate	(date) The date when the authentication record was created in WAS, in UTC date/time format.

updatedDate	(date) The date when the authentication record was updated in WAS, in UTC date/time format.
lastScan.date	(date) The date when the web application (associated with the authentication record) was last scanned, in UTC date/time format.
lastScan.authStatus	(keyword) Authentication status reported by the last web application scan: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL
isUsed	(boolean) Indicates whether used by a web application or scan.
contents	(Keyword: FORM_STANDARD, FORM_CUSTOM, FORM_SELENIUM, SERVER_BASIC, SERVER_DIGEST, SERVER_NTLM, CERTIFICATE, OAUTH2_AUTH_CODE, OAUTH2_IMPLICIT, OAUTH2_PASSWORD, and OAUTH2_CLIENT_CREDS)

Sample - Delete a single authentication record

Let us delete authentication record ID 78149.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/webappauthrecord
/78149"
```

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/3.
0/was/webappauthrecord.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WebAppAuthRecord>
      <id>78149</id>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```

Sample - Delete multiple authentication records

Let us delete authentication records that have a name containing the term “server”.

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">server</Criteria>
    </filters>
</ServiceRequest>
```

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/3.
0/was/webappauthrecord.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <data>
    <WebAppAuthRecord>
      <id>12874</id>
    <WebAppAuthRecord>
    <WebAppAuthRecord>
      <id>13093</id>
    <WebAppAuthRecord>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/webappauthrecord.xsd

Reference: Authentication

The <WebAppAuthRecord> element includes sub elements used to define authentication record. A reference of these elements is provided below. An asterisk * indicates a complex element.

Parameter	Description
id	(integer) Authentication record ID.
name	(text) Authentication record name.
tags	(integer) Tag associated with the authentication record.
tags.name	(text) Tag name assigned to the authentication record.
tags.id	(integer) Tag ID assigned to the authentication record.
createdDate	(date) The date when the authentication record was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the authentication record was updated in WAS, in UTC date/time format.
lastScan.date	(date) The date when the web application (associated with the authentication record) was last scanned, in UTC date/time format.
lastScan.authStatus	(keyword) Authentication status reported by the last web application scan: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL
isUsed	(boolean) Indicates whether used by a web application or scan.
contents	(Keyword: FORM_STANDARD, FORM_CUSTOM, FORM_SELENIUM, SERVER_BASIC,

	SERVER_DIGEST, SERVER_NTLM, CERTIFICATE, OAUTH2_AUTH_CODE, OAUTH2_IMPLICIT, OAUTH2_PASSWORD, and OAUTH2_CLIENT_CREDS)
WebAppAuthRecord	(text) Details associated with the web application authentication record. Use these parameters to create/update OAuth2 authentication record: WebAppAuthRecord.oauth2Record.grantType - (Required if authentication type is OAuth2)(text) Valid values are: 1) NONE, AUTH_CODE, IMPLICIT, PASSWORD, and CLIENT_CREDS. NONE means no grant type is selected. These are fields we support for each grant type: 1) AUTH_CODE - We support these fields for Authorization Code: 1) seleniumScript, 2) redirectUrl, 3) accessTokenUrl, 4) clientId (optional), 5) clientSecret (optional), 6) scope, (optional) and 7) accessTokenExpiredMsgPattern (optional) **Note:** Selenium script is mandatory for Authorization Code. We support parametrized username and password in the selenium script. See “Create a Selenium script to parameterize username and password” in the WAS API guide. 2) IMPLICIT - We support these fields for Implicit: 1) seleniumScript, and 2) redirectUrl **Note:** Selenium script is mandatory for Implicit. We support parametrized username and password in the selenium script. See “Create a Selenium script to parameterize username and password” in the WAS API guide. 3) PASSWORD - We support these fields for Resource Owner Password Credentials: 1)

	accessTokenUrl, 2) username, 3) password, 4) clientId (optional), 5) clientSecret (optional), 6) scope (optional), and 7) accessTokenExpiredMsgPattern (optional) 4) CLIENT_CREDS - We support these fields for Client Credentials: 1) accessTokenUrl, 2) clientId (optional), 3) clientSecret (optional), and 4) scope, (optional) Note: When creating an authentication record, you can specify either a Form record (used for web application authentication) or an OAuth2 record (used for the Swagger/Open API file authentication) in the request. While updating an authentication record, - Send the Form record with type as NONE if you want to set an OAuth2 record instead of a form record. - Send OAuth2 with grant type as NONE if you want to set a Form record instead of an OAuth2 record.
comments	(text) User-defined comments.

Catalog

Catalog Entry Count

/qps/rest/3.0/count/was/catalog

[GET] [POST]

Returns the total number of catalog entries in the user’s scope.

Permissions required - You must have the WAS module enabled. You must have the "API access" and "Access WAS module" permissions. You must have the "WAS.CATALOG.ACCESS" permission.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) The ID of the catalog entry.
ipAddress	(integer) The IP address of the discovered host. We support wild card character * for numbers in IP Address. For example, 10.11.196.* or 10.11.*.* are valid patterns for IP address. Examples of Invalid patterns:*1.123.123.123, 1*1.123.123.123 and 1*.123.123.123
port	(integer) The port number of the discovered service.
source	(text) The source of the catalog entries. Valid values are: VM_SCAN, VM_MAP, and WAS_SCAN.

Status	(text) The status of the entry. Valid values are NEW, ROGUE, APPROVED, IGNORED, IN_SUBSCRIPTION.
operatingSystem	(text) The operating system of discovered host.
netbiosName	(text) The NetBIOS name of the discovered host.
fqdn	(text) The fully qualified domain name of the discovered host.
createdDate	(date) The date and time when the catalog entry is created. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z
UpdatedDate	(date) The updated date and time when the catalog entry is updated. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z.

Sample - Get count of catalog entries (no criteria)

Returns the number (count) of all catalog entries in the user’s scope.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/catalog"
```

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/3.
0/was/catalog.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1355</count>
</ServiceResponse>
```

Search for a Catalog Entry

qps/rest/3.0/search/was/catalog

[POST]

Returns a list of catalog entries based on the search criteria.

Permissions required - You must have the WAS module enabled. You must have the "API access" and "Access WAS module" permissions. You must have the "WAS.CATALOG.ACCESS" permission.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) The ID of the catalog entry.
ipAddress	(integer) The IP address of the discovered host. We support wild card character * for numbers in IP Address. For example, 10.11.196.* or 10.11.*.* are valid patterns for IP address. Examples of Invalid patterns:*1.123.123.123, 1*1.123.123.123 and 1*.123.123.123
port	(integer) The port number of the discovered service.
source	(text) The source of the catalog entries. Valid values are: VM_SCAN, VM_MAP, and WAS_SCAN.
Status	(text) The status of the entry. Valid values are NEW, ROGUE, APPROVED, IGNORED, IN_SUBSCRIPTION.

operatingSystem	(text) The operating system of discovered host.
netbiosName	(text) The NetBIOS name of the discovered host.
fqdn	(text) The fully qualified domain name of the discovered host.
createdDate	(date) The date and time when the catalog entry is created. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z
UpdatedDate	(date) The updated date and time when the catalog entry is updated. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z.

Sample - Search for catalog entries

Let us view all catalog entries in the user’s scope for IP address that contains wild card character .

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
        <Criteria field="ipAddress"
          operator="EQUALS">10.113.*.*</Criteria>
    </filters>
</ServiceRequest>
```

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/3.
0/was/catalog.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <Catalog>
            <id>306909</id>
            <ipAddress>10.113.196.192</ipAddress>
            <port>443</port>
            <operatingSystem>Ubuntu / Fedora / Tiny Core Linux / Linux
             3.x</operatingSystem>
            <source>VM_SCAN</source>
            <status>ROGUE</status>
            <createdDate>2018-05-18T10:33:55Z</createdDate>
            <updatedDate>2020-05-19T13:50:08Z</updatedDate>
            <updatedBy>
                <id>1918433</id>
                <username>qualys_joe</username>
                <firstName>
                    <![CDATA[qualys]]>
                </firstName>
                <lastName>
                    <![CDATA[joe]]>
                </lastName>
            </updatedBy>
        </Catalog>
        <Catalog>
            <id>306906</id>
            <ipAddress>10.113.196.18</ipAddress>
            <port>80</port>
            <operatingSystem>Windows XP Service Pack
2</operatingSystem>
            <source>VM_SCAN</source>
            <fqdn>10-113-196-18.bogus.tld</fqdn>
            <netbiosName>SYS_10_113_196_18</netbiosName>
            <status>ROGUE</status>
            <createdDate>2018-05-18T10:33:55Z</createdDate>
            <updatedDate>2020-05-19T13:50:08Z</updatedDate>
            <updatedBy>
                <id>1918433</id>
                <username>qualys_joe</username>
                <firstName>
                    <![CDATA[qualys]]>
                </firstName>
                <lastName>
                    <![CDATA[joe]]>
                </lastName>
            </updatedBy>
        </Catalog>
    </data>
</ServiceResponse>
```

Get Catalog Entry Details

/qps/rest/3.0/get/was/catalog/{id}

[GET]

View the details of a catalog entry that is in your scope. In the output, “Comment” tag will show the comment added by the system and comment added by you.

Permissions required - You must have the WAS module enabled. You must have the "API access" and "Access WAS module" permissions. You must have the "WAS.CATALOG.ACCESS" permission.

Input Parameters

The element “id” (integer) is required, where “id” identifies the catalog entry.

Sample - View details of a catalog entry

Let us view details for the catalog entry with the ID 306904..

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/catalog/306904"
```

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/3.
0/was/catalog.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Catalog>
            <id>306904</id>
            <ipAddress>10.113.196.17</ipAddress>
            <port>80</port>
            <operatingSystem>MacOS X 9.0.0</operatingSystem>
            <source>VM_SCAN</source>
            <fqdn>10-113-196-17.bogus.tld</fqdn>
            <netbiosName>SYS_10_113_196_17</netbiosName>
            <status>NEW</status>
            <comments>
                <count>4</count>
                <list>
                    <Comment>
                        <contents>
                            <![CDATA[Web Application added from scan
consolidated data from VM]]>
                        </contents>
                        <createdDate>2018-05-
18T10:33:55Z</createdDate>
                    </Comment>
                    <Comment>
                        <contents>
                            <![CDATA[asdasd]]>
                        </contents>
                        <author>
                            <id>1918433</id>
                            <username>qualys_joe</username>
                            <firstName>
                                <![CDATA[qualys]]>
                            </firstName>
                            <lastName>
                                <![CDATA[joe]]>
                            </lastName>
                        </author>
                        <createdDate>2020-10-
22T07:47:25Z</createdDate>
                    </Comment>
                    <Comment>
                        <contents>
                            <![CDATA[Entry added to subscription as
'Catalog Web Application: 10-113-196-17.bogus.tld, Port 80']]>
                        </contents>
                        <createdDate>2020-10-
12T10:16:45Z</createdDate>
                    </Comment>
                </list>
            </comments>
            <createdDate>2018-05-18T10:33:55Z</createdDate>
            <updatedDate>2020-10-22T07:47:25Z</updatedDate>
            <updatedBy>
                <id>1918433</id>
                <username>qualys_joe</username>
                <firstName>
                    <![CDATA[qualys]]>
                </firstName>
                <lastName>
                    <![CDATA[joe]]>
                </lastName>
            </updatedBy>
        </Catalog>
    </data>
</ServiceResponse>
```

Update Catalog Entry

qps/rest/3.0/update/was/catalog/{id}

[POST]

Updates the status and comments for a catalog entry which is in your scope. Want to find an ID of a catalog entry to use as input? See Search catalog entries.

Permissions required - You must have the WAS module enabled. You must have the "API access" and "Access WAS module" permissions. You must have the "WAS.CATALOG.ACCESS" and "WAS.CATALOG.ENTRY.UPDATE" permissions.

Input Parameters

Parameter	Description
id	(integer) The element “id” is required, where “id” identifies a catalog entry.
status	(text) This is an optional parameter. The status can be updated to one of these statuses: ROGUE, NEW, APPROVED and IGNORED. IN_SUBSCRIPTION status can not be updated using the Update API.
Comments	(text) This is an optional parameter. You can add comments but you can not update/delete existing comments.

Sample - Search for catalog entries

Let us view all catalog entries in the user’s scope for IP address that contains wild card character .

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
```

```
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/catalog/368106"
<
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <data>
        <Catalog>
            <status>ROGUE</status>
            <comments>
                <add>
                    <Comment>
                        <contents>
                            <![CDATA[Comment 1]]>
                        </contents>
                    </Comment>
                </add>
            </comments>
        </Catalog>
    </data>
</ServiceRequest>
```

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/3.
0/was/catalog.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Catalog>
            <id>368106</id>
        </Catalog>
    </data>
</ServiceResponse>
```

Delete Catalog Entry

/qps/rest/3.0/delete/was/catalog/{id}

[POST]

Deletes a catalog entry which is in your scope. Want to find an ID of a catalog entry to use as input? See Search catalog entries.

Permissions required - You must have the WAS module enabled. You must have the "API access" and "Access WAS module" permissions. You must have the "WAS.CATALOG.ACCESS" and "WAS.CATALOG.ENTRY.DELETE" permissions.

Input Parameters

The element “id” (integer) is required, where “id” identifies a catalog entry.

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) The ID of the catalog entry.
ipAddress	(integer) The IP address of the discovered host. We support wild card character * for numbers in IP Address. For example, 10.11.196.* or 10.11.*.* are valid patterns for IP address. Examples of Invalid patterns:*1.123.123.123, 1*1.123.123.123 and 1*.123.123.123
port	(integer) The port number of the discovered service.
source	(text) The source of the catalog entries. Valid values are: VM_SCAN, VM_MAP, and WAS_SCAN.

Status	(text) The status of the entry. Valid values are NEW, ROGUE, APPROVED, IGNORED, IN_SUBSCRIPTION.
operatingSystem	(text) The operating system of discovered host.
netbiosName	(text) The NetBIOS name of the discovered host.
fqdn	(text) The fully qualified domain name of the discovered host.
createdDate	(date) The date and time when the catalog entry is created. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z
UpdatedDate	(date) The updated date and time when the catalog entry is updated. The date format is YYYY-MM-DDTHH:MM:SSZ. For example: 2018-05-18T10:33:54Z.

Sample - Delete a catalog entry

Let us delete a catalog entry with ID 368106.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/catalog/368106"
```

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/3.
0/was/catalog.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Catalog>
            <id>368106</id>
        </Catalog>
    </data>
</ServiceResponse>
```

Sample - Delete bulk catalog entries

Let us delete catalog entries in the user’s account with the given IDs.

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
        <Criteria field="id" operator="IN">610107,610110</Criteria>
    </filters>
</ServiceRequest>
```

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/3.
0/was/catalog.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <Catalog>
            <id>610107</id>
        </Catalog>
        <Catalog>
            <id>610110</id>
        </Catalog>
    </data>
</ServiceResponse>
```

Update Entries in Catalog

/qps/rest/3.0/updateEntries/was/catalog

[POST]

Updates the entries in the catalog to add data discovered in the most recent VM scan results within your account.

Permissions required - You must have the WAS module enabled. You must have the "API access" and "Access WAS module" permissions. You must have the "WAS.CATALOG.ACCESS" and "WAS.CATALOG.UPDATE" permissions.

Sample - Update entries in the catalog

Let us delete a catalog entry with ID 368106.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/updateEntries/was/catalog"
```

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/3.
0/was/catalog.xsd">
    <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

Add to Subscription

/qps/rest/3.0/addToSubscription/was/catalog/{id}

[POST]

Adds a web application entry to subscription to create a web application.

Permissions required - You must have the WAS module enabled. You must have the "API access" and "Access WAS module" permissions. You must have the "WAS.CATALOG.ACCESS" and "WAS.CATALOG.ENTRY.ADD_TO_SUBSCRIPTION" permissions.

Input Parameters

The element “id” (integer) is required, where “id” identifies the catalog entry.

Sample - Add a catalog entry to subscription

Let us add the catalog entry with id 306904 to subscription..

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/addToSubscription/was/catal
og/306904"
```

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/ve
rsion.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <success>
            <ids>413904,413906</ids>
            <count>1</count>
        </success>
        <duplicate>
```

```
            <count>1</count>
            <ids>413905</ids>
        </duplicate>
        <error>
            <count>2</count>
              <errorMessage>Invalid URL for web application catalog
              entries: 413907Some error occurred for web application
              catalog entries:413908
             </errorMessage>
        </error>
    </data>
</ServiceResponse>
```

Scans

Scan Count

/qps/rest/3.0/count/was/wasscan

[GET] [POST]

Returns the total number of scans in the user's account. Input elements are optional and are used to filter the number of scans included in the count.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The count includes scans in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) The scan ID.
name	(text) The scan name.
webApp.name	(text) The name of the web application being scanned.
webApp.id	(integer) The ID of the web application being scanned.

webApp.tags (with operator="NONE")	(integer) The tags associated with the web application being scanned.
webApp.tags.id	(integer) The tag ID assigned to web application being scanned.
reference	(text) Scan Reference ID.
launchedDate	(date) The date and time when the scan was launched in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).
type	(keyword) The scan type: VULNERABILITY or DISCOVERY.
mode	(keyword) The mode of the scan: ONDEMAND, SCHEDULED or API.
status	(keyword) The status of the scan: SUBMITTED, RUNNING, FINISHED, ERROR, CANCELED, PROCESSING.
authStatus	(Keyword) Indicates the status of the authentication record: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL.
resultsStatus	(keyword) The status of the scan: NOT_USED, TO_BE_PROCESSED, NO_HOST_ALIVE, NO_WEB_SERVICE, SERVICE_ERROR, TIME_LIMIT_REACHED, SCAN_INTERNAL_ERROR, SCAN_RESULTS_INVALID, SUCCESSFUL, PROCESSING, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, SUBMITTED, RUNNING, FINISHED, CANCELED, CANCELING, ERROR, DELETED, CANCELED_WITH_RESULTS.

Sample - Get count of scans in user's account

Return a count of all scans in the user's account.

API request

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscan"
```

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/3.
0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>534</count>
</ServiceResponse>
```

Sample - Get count of scans with certain criteria

Return a count of scans that match all the criteria defined in the request POST data: 1) scan name contains the word “Schedule”, 2) scan type is “VULNERABILITY”, 3) the scanned web application contains the word “Merchant”, and 4) the scan status is equal to “FINISHED”.

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
      <Criteria field="name" operator="CONTAINS">Schedule</Criteria>
      <Criteria field="type" operator="EQUALS">VULNERABILITY</Criteria
      <Criteria field="webApp.name"
operator="CONTAINS">Merchant</Criteria>
      <Criteria field="status" operator="EQUALS">FINISHED</Criteria>
    </filters>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```

Sample - Get the count of scans of web applications without tags

Return a count of scans of web applications that do not have any tags assigned.

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags" operator="NONE"></Criteria>
  </filters>
</ServiceRequest>
```

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/3.0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
</ServiceResponse>
```

Sample - Get the count of scans of web applications with few tags

Return a count of scans of web applications that have certain tags assigned.

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags.id"
operator="EQUALS">1516928</Criteria>
    <Criteria field="webApp.tags.id"
operator="EQUALS">1234567</Criteria>
  </filters>
</ServiceRequest>
```

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/3.0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>15</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscan.xsd

Search Scans

/qps/rest/3.0/search/was/wasscan

[POST]

Returns a list of scans on web applications which are in the user's scope

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes scans in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements

The special field=attributes attribute for the Criteria element is used to search custom attributes (see sample below).

Click here for available operators

Parameter	Description
id	(integer) The scan ID.
name	(text) The scan name.
webApp.name	(text) The name of the web application being scanned.
webApp.id	(integer) The ID of the web application being scanned.
webApp.tags (with operator="NONE")	(integer) The tags associated with the web application being scanned.

webApp.tags.id	(integer) The tag ID assigned to web application being scanned.
reference	(text) Scan Reference ID.
launchedDate	(date) The date and time when the scan was launched in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).
type	(keyword) The scan type: VULNERABILITY or DISCOVERY.
mode	(keyword) The mode of the scan: ONDEMAND, SCHEDULED or API.
status	(keyword) The status of the scan: SUBMITTED, RUNNING, FINISHED, ERROR, CANCELED, PROCESSING.
authStatus	(keyword) Indicates the status of the authentication record: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL.
resultsStatus	(keyword) The status of the scan: NOT_USED, TO_BE_PROCESSED, NO_HOST_ALIVE, NO_WEB_SERVICE, SERVICE_ERROR, TIME_LIMIT_REACHED, SCAN_INTERNAL_ERROR, SCAN_RESULTS_INVALID, SUCCESSFUL, PROCESSING, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, SUBMITTED, RUNNING, FINISHED, CANCELED, CANCELING, ERROR, DELETED, CANCELED_WITH_RESULTS.

Sample - List running scans

Let us view a list of all running scans in the user’s account.

API request

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

Request POST data

```
<ServiceRequest>
   <filters>
       <Criteria field="status" operator="EQUALS">RUNNING</Criteria>
    </filters>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.cm/qps/xsd/3.0
/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WasScan>
      <id>13101</id>
      <name><![CDATA[Vulnerability Scan - 2017-02-24]]></name>
      <reference>was/1298538355659.20994</reference>
      <type>VULNERABILITY</type>
      <mode>ONDEMAND</mode>
      <profile>
        <id>1072</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <launchedDate>2017-02-24T10:05:55Z</launchedDate>
      <launchedBy>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </launchedBy>
      <status>RUNNING</status>
      <consolidatedStatus>RUNNING</consolidatedStatus>
    </WasScan>
    <WasScan>
```

```
      <id>13102</id>
      <name><![CDATA[Vulnerability Scan - 2017-02-24]]></name>
      <reference>was/1298541157873.20995</reference>
      <type>VULNERABILITY</type>
      <mode>ONDEMAND</mode>
      <profile>
        <id>1072</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <launchedDate>2017-02-24T10:52:37Z</launchedDate>
      <launchedBy>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </launchedBy>
      <status>RUNNING</status>
      <consolidatedStatus>RUNNING</consolidatedStatus>
    </WasScan>
  </data>
</ServiceResponse>
```

Sample - List scans with successful authentication

Let us view a list of scans in the user’s account that successfully authenticated to the target web application.

API request

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

Request POST data

```
<ServiceRequest>
   <filters>
       <Criteria field="authStatus"
operator="EQUALS">SUCCESSFUL</Criteria>
   </filters>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WasScan>
      <id>13096</id>
      <name><![CDATA[Web Vulnerability Scan - 2017-02-23]]></name>
      <reference>was/1298475533625.20931</reference>
      <type>VULNERABILITY</type>
      <mode>ONDEMAND</mode>
      <profile>
        <id>1072</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <launchedDate>2017-02-23T16:38:53Z</launchedDate>
      <launchedBy>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </launchedBy>
      <status>FINISHED</status>
     <consolidatedStatus>NO_HOST_ALIVE</consolidatedStatus>
    </WasScan>
    <WasScan>
      <id>13116</id>
      <name><![CDATA[Relaunch Vulnerability Scan - 2017-02-
23]]></name>
      <reference>was/1298558684177.21009</reference>
      <type>VULNERABILITY</type>
      <mode>ONDEMAND</mode>
      <profile>
        <id>1072</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <launchedDate>2017-02-24T15:44:44Z</launchedDate>
      <launchedBy>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </launchedBy>
      <status>FINISHED</status>
      <consolidatedStatus>NO_HOST_ALIVE</consolidatedStatus>
    </WasScan>
  </data>
</ServiceResponse>
```

Sample - List scans for web applications without tags

Return a list of scans of web applications that do not have any tags assigned.

API request

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

Request POST data

```
<ServiceRequest>
   <filters>
        <Criteria field="webApp.tags" operator="NONE"></Criteria>
   </filters>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <WasScan>
            <id>2208317</id>
            <name>
                <![CDATA[1538976557822_Scan16]]>
            </name>
            <reference>was/1538976670564.372113</reference>
            <type>VULNERABILITY</type>
            <mode>API</mode>
            <multi>false</multi>
            <target>
                <webApp>
                    <id>1472824</id>
                    <name>
                        <![CDATA[web app 1538976530195]]>
                    </name>
                    <url>
                        <![CDATA[http://10.11.72.39]]>
                    </url>
                </webApp>
                <scannerAppliance>
                    <type>INTERNAL</type>
                    <friendlyName>
                        <![CDATA[John_doe]]>
                    </friendlyName>
                </scannerAppliance>
                <cancelOption>SPECIFIC</cancelOption>
                <randomizeScan>false</randomizeScan>
            </target>
            <profile>
                <id>458470</id>
                <name>
                    <![CDATA[My Option Profile - with defaults
1538976530177]]>
                </name>
            </profile>
            <launchedDate>2018-10-08T05:31:10Z</launchedDate>
            <launchedBy>
                <id>406790</id>
                <username>user_john</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Doe]]>
                </lastName>
            </launchedBy>
            <status>SUBMITTED</status>
              <consolidatedStatus>SUBMITTED</consolidatedStatus>
        </WasScan>
    </data>
</ServiceResponse>
```

Sample - List scans for web applications with tags

Return a list of scans of web applications that have certain tags assigned.

API request

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

Request POST data

```
<ServiceRequest>
   <filters>
        <Criteria field="webApp.tags.id"
operator="EQUALS">8158322</Criteria>
   </filters>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <WasScan>
            <id>2208317</id>
            <name>
                <![CDATA[1538976557822_Scan16]]>
            </name>
            <reference>was/1538976670564.372113</reference>
            <type>VULNERABILITY</type>
            <mode>API</mode>
            <multi>false</multi>
            <target>
                <webApp>
                    <id>1472824</id>
                    <name>
                        <![CDATA[web app 1538976530195]]>
                    </name>
                    <url>
                        <![CDATA[http://10.11.72.39]]>
                    </url>
                </webApp>
                <scannerAppliance>
                    <type>INTERNAL</type>
                    <friendlyName>
                        <![CDATA[John_doe]]>
                    </friendlyName>
                </scannerAppliance>
                <cancelOption>SPECIFIC</cancelOption>
                <randomizeScan>false</randomizeScan>
            </target>
            <profile>
                <id>458470</id>
                <name>
                    <![CDATA[My Option Profile - with defaults
1538976530177]]>
                </name>
            </profile>
            <launchedDate>2018-10-08T05:31:10Z</launchedDate>
            <launchedBy>
                <id>406790</id>
                <username>user_john</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Doe]]>
                </lastName>
            </launchedBy>
            <status>SUBMITTED</status>
            <consolidatedStatus>SUBMITTED</consolidatedStatus>
        </WasScan>
    </data>
</ServiceResponse>
```

Sample - List canceled scan

Let us search for the scan with response showing user who canceled the scan.

API request

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

Request POST data

```
<ServiceRequest>
   <filters>
         <Criteria field="id" operator="IN">1447989</Criteria>
   </filters>
</ServiceRequest>
```

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/3.
0/was/scan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <WasScan>
            <id>1447989</id>
            <name>
                <![CDATA[My Vulnerability Scan]]>
            </name>
            <reference>was/1446408743390.1856849</reference>
            <type>VULNERABILITY</type>
            <mode>ONDEMAND</mode>
            <multi>false</multi>
            <target>
                <webApp>
                    <id>2431279</id>
                    <name>
                        <![CDATA[127.0.0.1]]>
                    </name>
                    <url>
                        <![CDATA[http://127.0.0.1/]]>
                    </url>
                </webApp>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <cancelOption>SPECIFIC</cancelOption>
            </target>
            <profile>
                <id>28147</id>
                <name>
                    <![CDATA[My Option Profile]]>
                </name>
            </profile>
            <launchedDate>2017-11-01T20:12:23Z</launchedDate>
            <launchedBy>
                <id>2226741</id>
                <username>user_ak1</username>
                <firstName>
                    <![CDATA[Amy]]>
                </firstName>
                <lastName>
                    <![CDATA[Kim]]>
                </lastName>
            </launchedBy>
            <status>CANCELED</status>
            <consolidatedStatus>CANCELED</consolidatedStatus>
            <cancelMode>USER</cancelMode>
            <canceledBy>
                <id>9872437571</id>
                <username>user_bb5</username>
            </canceledBy>
        </WasScan>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscan.xsd

Get Scan Details

/qps/rest/3.0/get/was/wasscan/<id>

[GET]

View details for a scan on a web application which is in the user’s scope. Want to find a scan ID to use as input? See Search scans.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”. The output includes authentication records in the user's scope.

Input Parameters

The element “id” (integer) is required, where “id” identifies the scan.

Click here for available operators

Sample - List scan details

Let us view details for the scan with the ID 1447989.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscan/1447989"
```

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/3.
0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScan>
            <id>1447989</id>
            <name>
                <![CDATA[My Vulnerability Scan]]>
            </name>
            <reference>was/1446408743390.1856849</reference>
            <type>VULNERABILITY</type>
            <mode>ONDEMAND</mode>
            <progressiveScanning>DISABLED</progressiveScanning>
            <multi>false</multi>
            <target>
                <webApp>
                    <id>2431279</id>
                    <name>
                        <![CDATA[127.0.0.1]]>
                    </name>
                    <url>
                        <![CDATA[http://127.0.0.1/]]>
                    </url>
                </webApp>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <cancelOption>SPECIFIC</cancelOption>
            </target>
            <profile>
                <id>28147</id>
                <name>
                    <![CDATA[My Option Profile]]>
                </name>
            </profile>
            <options>
                <count>15</count>
                <list>
                    <WasScanOption>
                        <name>My Authentication Record</name>
                        <value>
                            <![CDATA[None]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Unexpected Error Threshold</name>
                        <value>
                            <![CDATA[48]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Sensitive Content: Credit Card
Numbers</name>
                        <value>
```

```
                            <![CDATA[false]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Performance Settings</name>
                        <value>
                            <![CDATA[MEDIUM]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Scanner Appliance</name>
                        <value>
                            <![CDATA[External]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Detection Scope</name>
                        <value>
                            <![CDATA[COMPLETE]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Crawling Form Submissions</name>
                        <value>
                            <![CDATA[NONE]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Bruteforce Settings</name>
                        <value>
                            <![CDATA[MINIMAL]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Option Profile Name</name>
                        <value>
                            <![CDATA[My Option Profile]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Maximum Crawling Links</name>
                        <value>
                            <![CDATA[300]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Timeout Error Threshold</name>
                        <value>
                            <![CDATA[20]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Web Application Name</name>
                        <value>
                            <![CDATA[127.0.0.1]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Request Parameter Set</name>
                        <value>
                            <![CDATA[Initial Parameters]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Sensitive Content: Social Security 
Numbers (US)</name>
                        <value>
                            <![CDATA[false]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Target URL</name>
                        <value>
                            <![CDATA[http://127.0.0.1/]]>
                        </value>
                    </WasScanOption>
                </list>
            </options>
            <launchedDate>2017-11-01T20:12:23Z</launchedDate>
            <launchedBy>
                <id>2226741</id>
                <username>user_ak1</username>
                <firstName>
                    <![CDATA[Amy]]>
                </firstName>
                <lastName>
                    <![CDATA[Kim]]>
                </lastName>
            </launchedBy>
            <status>CANCELED</status>
            <consolidatedStatus>CANCELED</consolidatedStatus>
            <cancelMode>USER</cancelMode>
            <canceledBy>
                <id>9872437571</id>
                <username>user_bb5</username>
            </canceledBy>
            <sendMail>true</sendMail>
            <sendOneMail>true</sendOneMail>
        </WasScan>
    </data>
</ServiceResponse>
```

Sample - List scan details with DNS override settings

When a scan has DNS override settings defined, the dnsOverride element lists DNS override settings (one or more records) used for scanning.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscan/1381602"
```

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/3.
0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScan>
            <id>1381602</id>
            <name>
                <![CDATA[My Scan]]>
            </name>
            <reference>was/1443153045656.1850463.1</reference>
            <type>DISCOVERY</type>
            <mode>ONDEMAND</mode>
            <multi>false</multi>
            <target>
                <webApp>
                    <id>1932867</id>
                    <name>
                        <![CDATA[10.10.10.2]]>
                    </name>
                    <url>
                        <![CDATA[http://10.10.10.2/]]>
                    </url>
                </webApp>
                <dnsOverride>
                    <id>1421</id>
                    <name>
                        <![CDATA[DNS Override Settings 1]]>
                    </name>
                </dnsOverride>
                <scannerAppliance>
>>>
```

Sample - Get details of a progressive scan

The progressiveScanning element will be included in the call response, if Progressive Scanning is enabled for the subscription. For all scans launched before this feature was enabled, the value “false” will be returned.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscan/31397"
```

XML response

```
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.
0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScan>
            <id>31397</id>
            <name>
                <![CDATA[Relaunch Relaunch Web Application
Vulnerability Scan - 2018-08-13]]>
            </name>
            <reference>was/1413891468597.1792880</reference>
            <type>VULNERABILITY</type>
            <mode>ONDEMAND</mode>
            <progressiveScanning>ENABLED</progressiveScanning>
...
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscan.xsd

Launch Scans (Single)

/qps/rest/3.0/launch/was/wasscan/

[POST]

We've enhanced the ability to support large web application scanning programs by adding the ability to scan any number of web applications as a Multi-Scan through API. This feature enables you to scan hundreds or even thousands of web applications you may have in your organization with granular insight into what scans are running and which ones are complete.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "Launch WAS Scan". The output includes scan targets in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. The special field=attributes attribute for the Criteria element is used to search custom attributes (see sample below).

Click here for available operators

Parameter	Description
name	(text) The scan name.
webApps.id or tags.id[1]	(integer) The web applications to be scanned. webApps.id: Specify the web application ID to include it in the scan. tags.id: Specify the tag ID associated with the web applications to be scanned.

type	(keyword) The scan type: VULNERABILITY or DISCOVERY.

profile.id$_2$	(integer) The name of the option profile that includes scan settings. The service provides the profile “Initial WAS Options” and we recommend this to get started.
	Example:
	`<profile>`
	`    <name>Initial WAS`
	`Options</name>`
	`</profile>`

target.scannerAppliance.type	(keyword) The type of scanner appliance used for the scan: EXTERNAL or INTERNAL or scannerTags.
target.scannerAppliance.friendlyName	(text) Name of the scanner appliance used for the scan.
target.scannerTags.set.Tag.id	(integer) The scanner associated with the tag (identified by the specified tag ID) is picked for the scan.
target.webAppAuthRecord.id or target.webAppAuthRecord.isDefault	Decides the authentication record to be used for the scan. target.webAppAuthRecord.id (integer): Specify the web application's authentication record ID to use the specific authentication record. target.webAppAuthRecord.isDefault (boolean): Set to true to use the default web application's authentication record for the scan.

proxy.id	(integer) The proxy for scanning the target web application.
	Example:
	`<proxy>`
	`   <id>12345</id>`
	`</proxy>`

dnsOverride.id	(integer) The DNS override record for scanning the target web application.
	Example:
	`<dnsOverride>`
	`   <id>67890</id>`
	`</dnsOverride>`

sendMail	(boolean) Set to false to disable scan complete email notifications. `Example:<sendMail>false</sendMail>`

1 The element target must have at least tags or web applications specified

2 The element profile (Text) is required unless the target has a default option profile.

Sample - Launch a new scan - basic elements

Launch a new discovery scan on the web application ID 323126 using the option profile ID 1021.

API request

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

Request POST data

```
<ServiceRequest>
  <data>
    <WasScan>
      <name>New WAS Discovery Scan launched from API</name>
      <type>DISCOVERY</type>
      <target>
        <webApp>
          <id>323126</id>
        </webApp>
        <webAppAuthRecord>
          <isDefault>true</isDefault>
        </webAppAuthRecord>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
      </target>
    <profile>
        <id>1021</id>
    </profile>
  </WasScan>
</data>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>16954</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Sample - Launch a new scan - use proxy

Launch a new vulnerability scan using proxy ID 12345.

API request

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <data>
    <WasScan>
      <name>New WAS Vulnerability Scan launched from API</name>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
          <id>323126</id>
        </webApp>
        <scannerAppliance>
          <type>INTERNAL</type>
          <friendlyName>dp_scanner</friendlyName>
        </scannerAppliance>
        <proxy>
          <id>12345</id>
        </proxy>
      </target>
      <profile>
        <id>1021</id>
      </profile>
    </WasScan>
  </data>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>16954</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Sample - Launch a new scan - assign multiple scanner appliances

Let us launch a new discovery scan on the web application ID 522066 and assign the pool of scanners using asset tag.

API request

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

Request POST data

```
<ServiceRequest>
   <data>
     <WasScan>
      <name><![CDATA[Scan With Pool of Internal Scanners]></name>
      <type>DISCOVERY</type>
      <target>
        <webApp>
           <id>522066</id>
        </webApp>
           <scannerTags>
             <set>
               <Tag>
                 <id>15415353311147</id>
               </Tag>
             </set>
           </scannerTags>
       </target>
     </WasScan>
   </data>
</ServiceRequest>
```

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/3.
0/was/scan.xsd">
```

```
<responseCode>SUCCESS</responseCode>
  <count>1</count>
<data>
        <WasScan>
            <id>1731352</id>
            <name><![CDATA[Scan With Pool of Internal 
Scanners]]></name>
            <reference>was/1484222839357.1955345</reference>
            <type>DISCOVERY</type>
            <mode>ONDEMAND</mode>
            <progressiveScanning>ENABLED</progressiveScanning>
            <multi>true</multi>
            <target>
                <webApps>
                    <list>
                        <WebApp>
                            <id>522066</id>
                            <name><![CDATA[My Web Application]]></name>
                            <url><![CDATA[http://mywebapp.com]]></url>
                        </WebApp>
            </list>
                </webApps>
                <scannerTags>
                    <set>
                        <Tag>
                            <id>8461819</id>
                            <name><![CDATA[TagForScanner]]></name>
                        </Tag>
                    </set>
                </scannerTags>
                <cancelOption>DEFAULT</cancelOption>
            </target>
            <profile>
                <id>194283</id>
                <name>
                    <![CDATA[Initial WAS Options]]>
                </name>
            </profile>
            <options>
                <count>14</count>
                <list>
                    <WasScanOption>
                        <name>Web Application Authentication Record 
Name</name>
                        <value><![CDATA[None]]></value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Unexpected Error Threshold</name>
                        <value>
                            <![CDATA[300]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Sensitive Content: Credit Card
Numbers</name>
                        <value>
                            <![CDATA[false]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Performance Settings</name>
                        <value>
                            <![CDATA[LOW]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Detection Scope</name>
                        <value>
                            <![CDATA[COMPLETE]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Crawling Form Submissions</name>
                        <value>
                            <![CDATA[BOTH]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Bruteforce Settings</name>
                        <value>
                            <![CDATA[DISABLED]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Option Profile Name</name>
                        <value>
                            <![CDATA[Initial WAS Options]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
```

```
                        <name>Maximum Crawling Links</name>
                        <value>
                            <![CDATA[300]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Timeout Error Threshold</name>
                        <value>
                            <![CDATA[100]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Web Application Name</name>
                        <value>
                            <![CDATA[My Web Application]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Request Parameter Set</name>
                        <value>
                            <![CDATA[Initial Parameters]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Sensitive Content: Social Security 
Numbers (US)</name>
                        <value>
                            <![CDATA[false]]>
                        </value>
                    </WasScanOption>
                    <WasScanOption>
                        <name>Target URL</name>
                        <value>
                            <![CDATA[http://mywebapp.com]]>
                        </value>
                    </WasScanOption>
                </list>
            </options>
            <launchedDate>2017-01-12T12:07:19Z</launchedDate>
            <launchedBy>
                <id>1056860</id>
                <username>user_john</username>
                 <firstName><![CDATA[John]]></firstName>
                 <lastName><![CDATA[Doe]]></lastName>
            </launchedBy>
            <status>SUBMITTED</status>
            <sendMail>true</sendMail>
        </WasScan>
    </data>
</ServiceResponse>
```

Sample - Launch a new scan - progressive scanning

The user can set the progressiveScanning option to true or false for the vulnerability scan, if Progressive Scanning is enabled for the subscription. If the option is not set for a scan, the Progressive Scanning setting for the web application is used. Note this option is not supported for a discovery scan.

API request

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

Request POST data

```
<ServiceRequest>
  <data>
    <WasScan>
      <name>New WAS Vulnerability Scan launched from API</name>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
            <id>323126</id>
        </webApp>
    <scannerAppliance>
       <type>EXTERNAL</type>
    </scannerAppliance>
      </target>
        <profile>
            <id>1021</id>
        </profile>
        <cancelAfterNHours>5</cancelAfterNHours>
         <progressiveScanning>DISABLED</progressiveScanning>
    </WasScan>
  </data>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>16954</id>
    </WasScan>
  </data>
</ServiceResponse>
```

If Progressive Scanning is not enabled for the subscription, the progressiveScanning element cannot be provided, otherwise an error will be returned.

XML response (error)

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.
0/was/wasscan.xsd">
    <responseCode>INVALID_REQUEST</responseCode>
    <responseErrorDetails>
        <errorMessage>Progressive scanning is not enabled in your
subscription.</errorMessage>
        <errorResolution>Please check with your account manager to
enable this option.</errorResolution>
    </responseErrorDetails>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscan.xsd

Launch Scan (Multiple)

/qps/rest/3.0/launch/was/wasscan

[POST]

We've enhanced the ability to support large web application scanning programs by adding the ability to scan any number of web applications as a Multi-Scan through API. This feature enables you to scan hundreds or even thousands of web applications you may have in your organization with granular insight into what scans are running and which ones are complete.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "Launch WAS Scan". The output includes scan targets in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements

The special field=attributes attribute for the Criteria element is used to search custom attributes (see sample below).

Click here for available operators

Parameter	Description
name	(text) The scan name.
target.webApp.id[1]	(integer) The ID of the web application being scanned.
target.tags.excluded.option	(keyword: ALL or ANY) Decides which web applications should be excluded from the scan.

	ALL : Only the web applications associated with all the specified tags are excluded from the scan. ANY : Only the web applications associated with any of the specified tags are excluded from the scan.
target.tags.excluded.tagList.Tag.id	(integer) The web applications associated with the tag (identified by the specified tag ID) are excluded from the scan.
target.tags.included.option	(keyword: ALL or ANY) Decides which web applications should be included in the scan. ALL : Only the web applications associated with all the specified tags are included in the scan. ANY : Only the web applications associated with any of the specified tags included in the scan.
target.tags.included.tagList.Tag.id	(integer) The web applications associated with the tag (identified by the specified tag ID) are included in the scan.
options	(keyword: ANY, ALL) Decides which web applications should be included or excluded from the scan. ALL : Only the web applications associated with all the specified tags are excluded from the scan. ANY : Only the web applications associated with any of the specified tags are excluded from the scan.
type	(keyword: EXTERNAL or INTERNAL or scannerTags) Type of the scanner appliance to be used for the scan.

profile.id$_2$	(integer) (integer) The name of the option profile that includes scan settings. The service provides the profile “Initial WAS Options” and we recommend this to get started.
	`Example:`
	`<profile>`
	`    <name>Initial WAS Options</name>`
	`</profile>`

target.authRecordOption	(integer) Defines the authentication record to be used during the scan. Set to SPECIFIC -Always use the authRecord passed while launching the scan. Set to DEFAULT- Forces the use of the authRecord, if set, else fall back to the one passed in to the API while launching the scan.
target.profileOption	(keyword: ALL or ANY) Defines the option profile to be used during the scan. Set to SPECIFIC - Always use the optionProfile passed while launching the scan. Set to DEFAULT - Forces the use of the optionProfile if set, else fall back to the one passed in to the API while launching the scan.
target.scannerOption	(integer) Defines the scanner appliance to be used during the scan. Set to SPECIFIC - Always use the scanner passed while launching the scan

	Set to DEFAULT - Forces the use of the scanner if set, else fall back to the one passed in to the API while launching the scan.
<cancelOption>	Set to DEFAULT - Forces the use of the target web application's cancelScans option if set, else fall back to the one passed in to the API while launching the scan. Set to SPECIFIC - Always use the cancel scan option passed while launching the scan.
sendMail	(boolean) Set to false to disable scan complete email notifications. `Example:<sendMail>false</sendMail>`
sendOneMail	(boolean) Set to true to send one email upon multi-scan completion. Set to false to send one email upon completion of each individual scan. `Example:<sendOneMail>true</sendOneMail>` Note: sendOneMail is valid only when sendMail = true for a multi-scan (multiple web applications being scanned). If sendMail is set to false, sendOneMail will be ignored.

1 The element target must have at least tags or web applications specified

2 The element profile (Text) is required unless the target has a default option profile.

Sample - Launch a new scan - basic elements

Launch a new discovery scan on the web application ID 4330527 and 4330538 using the option profile ID 1070535.

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <WasScan>
            <name>1497343127459_Scan7</name>
            <type>DISCOVERY</type>
            <target>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <webApps>
                    <set>
                        <WebApp>
                            <id>4330527</id>
                        </WebApp>
                        <WebApp>
                            <id>4330338</id>
                        </WebApp>
                    </set>
                </webApps>
                <profileOption>DEFAULT</profileOption>
            </target>
            <profile>
                <id>1070535</id>
            </profile>
        </WasScan>
    </data>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScan>
            <id>2281862</id>
        </WasScan>
    </data>
</ServiceResponse>
```

Sample - Launch a multi-scan using tags

Let’s launch a multi- scan for all the web applications associated with the tags specified in the request filter.

API request

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <WasScan>
            <name>1497343127649_Scan9</name>
            <type>DISCOVERY</type>
            <target>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <tags>
                    <included>
                        <option>ALL</option>
                        <tagList>
                            <set>
                                <Tag><id>12017424</id></Tag>
                                <Tag><id>12017228</id></Tag>
                            </set>
                        </tagList>
                    </included>
                    <excluded>
                        <option>ANY</option>
                        <tagList>
```

```
                            <set>
                                <Tag>
                                    <id>12017228</id>
                                </Tag>
                            </set>
                        </tagList>
                    </excluded>
                </tags>
                <scannerOption>DEFAULT</scannerOption>
            </target>
            <profile>
                <id>1070535</id>
            </profile>
        </WasScan>
    </data>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScan>
            <id>2281863</id>
        </WasScan>
    </data>
</ServiceResponse>
```

Sample - Launch a new scan with cancel option to DEFAULT

Launch a new vulnerability scan on web app ID 2376280 and 4114251 and set the cancel scan option to DEFAULT. This forces the use of the target web app’s cancelScans option if set.

API request

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

Request POST data

```
<ServiceRequest>
<data>
<WasScan>
     <name><![CDATA[sample Scan]]></name>
       <type>VULNERABILITY</type>
         <target>
          <webApps>
           <set>
            <WebApp>
              <id>2376280</id>
            </WebApp>
           <WebApp>
             <id>4114251</id>
           </WebApp>
          </set>
        </webApps>
      <scannerAppliance>
          <type>EXTERNAL</type>
      </scannerAppliance>
     <cancelOption>DEFAULT</cancelOption>
      </target>
     <profile>
      <id>2231014</id>
     </profile>
  </WasScan>
 </data>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>1275177</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Sample - Launch a new multi-scan

Let us launch a scan that allows to send one email on completion of multi-scan (not for each individual scan in the group).

API request

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

Request POST data

```
<ServiceRequest>
   <data>
        <WasScan>
        <name><![CDATA[New Scan]]></name>
            <type>VULNERABILITY</type>
            <target>
    <webApps>
   <set>
            <WebApp><id>8389207</id></WebApp>
            <WebApp><id>8389244</id></WebApp>
   </set>
   </webApps>
     <scannerAppliance>
            <type>EXTERNAL</type>
     </scannerAppliance>
     </target>
     <profile>
          <id>2337683</id>
     </profile>
           <sendOneMail>true</sendOneMail>
      </WasScan>
    </data>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xs
d/3.0/was/wasscan.xsd">
```

```
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>3456140</id>
    </WasScan>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscan.xsd

Scan Again

/qps/rest/3.0/scanagain/was/scan/<id>

[POST]

We now provide the option to execute a previous scan again. Identify the scan you want to run again and use scanagain action. We'll do our best to pre-fill the scan settings to match the original scan.

Permissions required - User must have WAS module enabled. User account must have these permissions: "API Access" and "Access WAS module". The web application must be in the user's scope.

Input Parameters

The element “id” (integer) is required, where “id” identifies the scan to be executed again. You could optionally provide a new name for the scan as well.

Click here for available operators

Sample - Scan with Scanagain option

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/scanagain/was/wasscan/46263
54"
```

Request POST data

```
<ServiceRequest>
  <data>
        <WasScan>
            <name>Sample Scan Name for Rescan</name>
        </WasScan>
    </data>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1 </count>
     <data>
        <WasScan>
            <id>4626354</id>
        </WasScan>
    </data>
</ServiceResponse>
```

Retrieve Scan Status

/qps/rest/3.0/status/was/wasscan/<id>

[GET]

Retrieve the status of a scan on a web application which is in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes scan targets in the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies the scan.

Click here for available operators

Sample - View scan status along with authentication status

View details for the scan with the ID 1902350.

API request

```
curl -n -u
"USERNAME:PASSWORD"  "https://qualysapi.qualys.com/qps/rest/3.0/status
/was/wasscan/1902350"
```

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/3.
0/was/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScan>
            <id>1902350</id>
            <status>FINISHED</status>
            <consolidatedStatus>NO_HOST_ALIVE</consolidatedStatus>
            <summary>
                <resultsStatus>NO_HOST_ALIVE</resultsStatus>
                <authStatus>NONE</authStatus>
            </summary>
        </WasScan>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/scan.xsd

Retrieve Scan Results

/qps/rest/3.0/download/was/wasscan/<id>

/qps/rest/2.0/download/was/wasscan/<id>

[GET]

Retrieve the results of a scan on a web application which is in the user's scope. Include "3.0" in the URL for WASA v3 scan results using the WAS API schema, part of the API V3 architecture (see https://qualysapi.qualys.com/qps/xsd/3.0/was/wasscan.xsd). Include "2.0" in the URL for scan results in legacy format (WAS v2 and earlier), using the webapp_scan.dtd - see Reference: WAS Scan Results (legacy).

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes scan targets in the user's scope.

The Retrieve Scan Results APIs can also generate scan response based on the detection groups.

Tip: When you download web application scan results using the WAS API, you'll want to view vulnerability descriptions from the Qualys KnowledgeBase in order to understand the vulnerabilities detected and see our recommended solutions. See How to Download Vulnerability Details.

Input Parameters

The element "id" (integer) is required, where "id" identifies the scan.

Click here for available operators

Sample - Download results of a scan

Download the results of the scan with the ID 174726.

API request

```
curl -n -u
"USERNAME:PASSWORD"  "https://qualysapi.qualys.com/qps/rest/3.0/downlo
ad/was/wasscan/174726"
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<WasScan xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.
0/was/wasscan.xsd">
  <id>174726</id>
  <name><![CDATA[My Web Application Scan]]></name>
  <reference>was/1328563860860.218807</reference>
  <type>VULNERABILITY</type>
  <mode>API</mode>
  <target>
    <webApp>
      <id>952835</id>
      <name><![CDATA[My Web Application]]></name>
      <url><![CDATA[https://example.com/]]></url>
    </webApp>
    <scannerAppliance>
      <type>INTERNAL</type>
      <friendlyName><![CDATA[is_joe_user]]></friendlyName>
    </scannerAppliance>
  </target>
  <profile>
    <id>6714</id>
    <name><![CDATA[Initial WAS Options]]></name>
  </profile>
  <options>
    <count>10</count>
    <list>
      <WasScanOption>
        <name>Detection Scope</name>
        <value>COMPLETE</value>
      </WasScanOption>
      <WasScanOption>
        <name>Maximum Crawling Links</name>
        <value>300</value>
      </WasScanOption>
      <WasScanOption>
        <name>Bruteforce Settings</name>
        <value>MINIMAL</value>
      </WasScanOption>
      <WasScanOption>
        <name>Option Profile Name</name>
        <value>Initial WAS Options</value>
      </WasScanOption>
<WasScanOption>
        <name>Scanner Appliance Name</name>
<value><![CDATA[External (IP: 10.40.3.104, Scanner: 6.2.13-1, WAS:
2.13.5-1, Signatures: 2.2.52-2)]]></value>
      </WasScanOption>
<WasScanOption>
   <name>Ignore Binary Files</name>
   <VALUE><![CDATA[true]]></VALUE>
</WasScanOption>
...
    </list>
  </options>
  <launchedDate>2017-02-06T21:31:00Z</launchedDate>
  <launchedBy>
    <id>35842</id>
    <username>joe_user</username>
    <firstName><![CDATA[John]]></firstName>
    <lastName><![CDATA[Smith]]></lastName>
  </launchedBy>
  <status>FINISHED</status>
  <consolidatedStatus>NO_HOST_ALIVE</consolidatedStatus>
  <endScanDate>2017-02-06T21:49:34Z</endScanDate>
  <scanDuration>1114</scanDuration>
  <summary>
    <crawlDuration>16</crawlDuration>
    <testDuration>138</testDuration>
    <linksCollected>10</linksCollected>
    <linksCrawled>1</linksCrawled>
    <nbRequests>503</nbRequests>
    <averageResponseTime>0.001554</averageResponseTime>
    <resultsStatus>SUCCESSFUL</resultsStatus>
    <authStatus>NONE</authStatus>
  </summary>
  <stats>
    <global>
      <nbVulnsTotal>79</nbVulnsTotal>
      <nbVulnsLevel5>24</nbVulnsLevel5>
      <nbVulnsLevel4>0</nbVulnsLevel4>
      <nbVulnsLevel3>3</nbVulnsLevel3>
      <nbVulnsLevel2>18</nbVulnsLevel2>
      <nbVulnsLevel1>34</nbVulnsLevel1>
      <nbScsTotal>0</nbScsTotal>
      <nbScsLevel5>0</nbScsLevel5>
      <nbScsLevel4>0</nbScsLevel4>
      <nbScsLevel3>0</nbScsLevel3>
      <nbScsLevel2>0</nbScsLevel2>
      <nbScsLevel1>0</nbScsLevel1>
      <nbIgsTotal>10</nbIgsTotal>
      <nbIgsLevel5>0</nbIgsLevel5>
      <nbIgsLevel4>0</nbIgsLevel4>
      <nbIgsLevel3>0</nbIgsLevel3>
      <nbIgsLevel2>0</nbIgsLevel2>
      <nbIgsLevel1>10</nbIgsLevel1>
    </global>
    <byGroup>
      <count>3</count>
      <list>
        <GroupStat>
          <group>PATH</group>
          <nbTotal>18</nbTotal>
          <nbLevel5>0</nbLevel5>
          <nbLevel4>0</nbLevel4>
          <nbLevel3>0</nbLevel3>
          <nbLevel2>18</nbLevel2>
          <nbLevel1>0</nbLevel1>
        </GroupStat>
...
      </list>
    </byGroup>
    <byOwasp>
      <count>4</count>
      <list>
        <OwaspStat>
          <owasp>OWASP-A4</owasp>
          <nbTotal>18</nbTotal>
          <nbLevel5>0</nbLevel5>
          <nbLevel4>0</nbLevel4>
          <nbLevel3>0</nbLevel3>
          <nbLevel2>18</nbLevel2>
          <nbLevel1>0</nbLevel1>
        </OwaspStat>
...
      </list>
    </byOwasp>
    <byWasc>
      <count>5</count>
      <list>
        <WascStat>
          <wasc>WASC-15</wasc>
          <nbTotal>14</nbTotal>
          <nbLevel5>0</nbLevel5>
          <nbLevel4>0</nbLevel4>
          <nbLevel3>2</nbLevel3>
          <nbLevel2>12</nbLevel2>
          <nbLevel1>0</nbLevel1>
        </WascStat>
...
      </list>
    </byWasc>
  </stats>
  <vulns>
    <count>79</count>
    <list>
      <WasScanVuln>
        <qid>150081</qid>
        <title><![CDATA[Possible Clickjacking vulnerability]]></title>
        <uri><![CDATA[https://example.com/randomLink/1328558353.9231]]
></uri>
        <instances>
          <count>1</count>
          <list>
            <WasScanVulnInstance>
              <authenticated>false</authenticated>
              <payloads>
                <count>1</count>
                <list>
                  <WasScanVulnPayload>
                    <payload><![CDATA["'>
<qss%20a=@REQUESTID@>]]></payload>
                    <result base64="true">
<![CDATA[c3RhcnQoKTogVGhlIHNlc3Npb24gaWQgY29udGFpbnMgaW52YWxpZCBjaGFyY
WN0ZXJzLCB2YWxpZCBjaGFyYWN0ZXJzIGFyZSBvbmx5IGEteiwgQS1aIGFuZCAwLTkgaW4
gJmx0O2ImZ3Q7L3Zhci93d3cvaHRtbC9pbmNsdWRlcy9jb25maWcucGhwJmx0Oy9iJmd0O
yBvbiBsaW5lICZsdDtiJmd0OzImbHQ7L2ImZ3Q7Jmx0O2JyIC8mZ3Q7CiZsdDticiAvJmd
0OwombHQ7YiZndDtXYXJuaW5nJmx0Oy9iJmd0OzogIHNlc3Npb25fc3RhcnQoKTogQ2Fub
m90IHNlbmQgc2Vzc2lvbiBjYWNoZSBsaW1pdGVyIC0gaGVhZGVycyBhbHJlYWR5IHNlbnQ
gKG91dHB1dCBzdGFydGVkIGF0IC92YXIvd3d3L2h0bWwvaW5jbHVkZXMvY29uZmlnLnBoc
DoyKSBpbiAmbHQ7YiZndDsvdmFyL3d3dy9odG1sL2luY2x1ZGVzL2NvbmZpZy5waHAmbHQ
7L2ImZ3Q7IG9uIGxpbmUgJmx0O2ImZ3Q7MiZsdDsvYiZndDsmbHQ7YnIgLyZndDsKJmx0O
2JyIC8mZ3Q7CiZsdDtiJmd0O1dhcm5pbmcmbHQ7L2ImZ3Q7OiAgQ2Fubm90IG1vZGlmeSB
oZWFkZXIgaW5mb3JtYXRpb24gLSBoZWFkZXJzIGFscmVhZHkgc2VudCBieSAob3V0cHV0I
HN0YXJ0ZWQgYXQgL3Zhci93d3cvaHRtbC8]]></result>
                  </WasScanVulnPayload>
                </list>
              </payloads>
            </WasScanVulnInstance>
          </list>
        </instances>
      </WasScanVuln>
...
    </list>
  </vulns>
  <sensitiveContents>
    <count>0</count>
  </sensitiveContents>
  <igs>
    <count>10</count>
    <list>
      <WasScanIg>
        <qid>150058</qid>
        <title><![CDATA[Flash Analysis]]></title>
        <data base64="true"><![CDATA
[U1dGIGZpbGU6IGh0dHA6Ly8xMC4xMC4yNi4yMzg6ODAvYm9xL2FjY3QvcGVyc29uYWwvd
2ludGVyMi5zd2YKICAgICBWZXJzaW9uOiA4CgpTV0YgZmlsZTogaHR0cDovLzEwLjEwLjI
2LjIzOC9ib3EvcHJvdGVjdGVkL21pbWUvZGVmYXVsdFBhZ2Uuc3dmCiAgICAgVmVyc2lvb
jogNgoK]]></data>
      </WasScanIg>
...
    </list>
  </igs>
</WasScan>
```

Sample - Download results of a scan with SSL/TLS details

API request

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

XML response

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

```
<WasScan xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/3.0
/was/wasscan.xsd">
    <id>3217161</id>
    <name>
        <![CDATA[[SSL-Certs]] 2020-01-30 6:20:49PM]]>
    </name>
    <reference>was/1580388655076.626241</reference>
    <type>VULNERABILITY</type>
    <mode>ONDEMAND</mode>
    <progressiveScanning>DISABLED</progressiveScanning>
    <multi>false</multi>
    <target>
        <webApp>
            <id>3016632</id>
            <name>
                <![CDATA[SSL-Certs]]>
            </name>
            <url>
                <![CDATA[https://10.115.78.72/welcome.html]]>
            </url>
        </webApp>
        <scannerAppliance>
            <type>INTERNAL</type>
            <friendlyName>
                <![CDATA[WAS_Scanner_vp1]]>
            </friendlyName>
        </scannerAppliance>
        <cancelOption>SPECIFIC</cancelOption>
    </target>
    <profile>
        <id>893488</id>
        <name>
            <![CDATA[ssl]]>
        </name>
    </profile>
    <options>
        <count>16</count>
        <list>
            <WasScanOption>
                <name>Web Application Authentication Record
Name</name>
                <value>
                    <![CDATA[None]]>
                </value>
            </WasScanOption>
...
<list>
            <WasScanIg>
                <qid>38704</qid>
                <title>
                    <![CDATA[SSL/TLS Key Exchange Methods]]>
                </title>
                <sslData>
...
<sslDataInfoList>
                        <list>
                            <SSLDataInfo>
                                <sslDataKexList>
                                    <list>
                                        <SSLDataKex>
                                            <protocol>TLSv1</protocol>
                                            <kex>ECDHE</kex>
                                            <group>x25519</group>
                                            <keysize>256</keysize>
                                            <fwdsec>yes</fwdsec>
                                            <classical>128</classical>
                                            <quantum>low</quantum>
                                        </SSLDataKex>
...
<WasScanIg>
                <qid>38706</qid>
                <title>
                    <![CDATA[SSL/TLS Protocol Properties]]>
                </title>
                <sslData>
...
<sslDataInfoList>
                        <list>
                            <SSLDataInfo>
                                <sslDataPropList>
                                    <list>
                                        <SSLDataProp>
                                            <name>Extended Master
Secret</name>
                                            <value>yes</value>
                                            <protocol>TLSv1</protocol>
                                        </SSLDataProp>
                                        <SSLDataProp>
```

```
                                            <name>Encrypt Then
MAC</name>
                                            <value>yes</value>
                                            <protocol>TLSv1</protocol>
                                        </SSLDataProp>
...
<WasScanIg>
                <qid>6</qid>
                <title>
                    <![CDATA[DNS Host Name]]>
                </title>
                <sslData>
...
<sslDataInfoList>
                        <list>
                            <SSLDataInfo>
                                <certificateFingerprint>291126AC8ED272
F71EDF06E5B76BBECD1C811769D4FE988DE95FF848AFEBCF6A</certificateFingerp
rint>
                            </SSLDataInfo>
                        </list>
                    </sslDataInfoList>
...
<WasScanIg>
                <qid>38291</qid>
                <title>
                    <![CDATA[SSL Session Caching Information]]>
                </title>
...
<WasScanIg>
                <qid>45017</qid>
                <title>
                    <![CDATA[Operating System Detected]]>
                </title>
                <sslData>
                    <protocol>tcp</protocol>
                    <ip>10.115.78.72</ip>
                    <port>0</port>
                    <result>
                        <![CDATA[Ubuntu_/_Fedora_/_Tiny_Core_Linux_/_L
inux_3.x TCP/IP_Fingerprint U5933:443
]]>
                    </result>
                </sslData>
...
```

```
<WasScanIg>
                <qid>38116</qid>
                <title>
                    <![CDATA[SSL Server Information Retrieval]]>
                </title>
                <sslData>
...
<sslDataInfoList>
                        <list>
                            <SSLDataInfo>
                                <sslDataCipherList>
                                    <list>
                                        <SSLDataCipher>
                                            <protocol>TLSv1</protocol>
                                            <name>ECDHE-RSA-AES128-
SHA</name>
                                            <keyExchange>ECDH</keyExch
ange>
                                            <auth>RSA</auth>
                                            <mac>SHA1</mac>
                                            <encryption>AES(128)</encr
yption>
                                            <grade>MEDIUM</grade>
                                        </SSLDataCipher>
...
 </igs>
    <sendMail>true</sendMail>
    <enableWAFAuth>false</enableWAFAuth>
</WasScan>
```

Sample - Download WAS Scan Details based on Detection Group

API request

```
curl -n -u "USERNAME:PASSWORD"
<qualys_base_url>/qps/rest/3.0/download/was/wasscan/7375164"
```

XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<WasScan
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="
    <qualys_base_url>/qps/xsd/3.0/was/wasscan.
xsd">
```

```
        <id>7375164</id>
        <name>
            <![CDATA[Web Application Vulnerability Scan - Copy of 
Reports - New
Web app1 - 2023-10-27 Run #156]]>
        </name>
        <reference>was/1712169015838.4088737</reference>
        <type>VULNERABILITY</type>
        <mode>SCHEDULED</mode>
        <progressiveScanning>ENABLED</progressiveScanning>
        <multi>false</multi>
        <target>
            <webApp>
                <id>36673912</id>
                <name>
                    <![CDATA[Copy of Reports - New Web app1]]>
                </name>
                <url>
                    <![CDATA[http://funkytown.vuln.qa.qualys.com]]>
                </url>
            </webApp>
            <scannerAppliance>
                <type>INTERNAL</type>
                <friendlyName>
                    <![CDATA[WAS_Scanner_rs1]]>
                </friendlyName>
            </scannerAppliance>
            <cancelOption>SPECIFIC</cancelOption>
        </target>
        <profile>
            <id>2237691</id>
            <name>
                <![CDATA[100 links]]>
            </name>
        </profile>
        <options>
            <count>16</count>
            <list>
                <WasScanOption>
                    <name>Web Application Authentication Record 
Name</name>
                    <value>
                        <![CDATA[None]]>
                    </value>
                </WasScanOption>
                <WasScanOption>
                    <name>Sensitive Content: Credit Card
Numbers</name>
                    <value>
                        <![CDATA[true]]>
                    </value>
                </WasScanOption>
            </list>
        </options>
.......
.......
.......
        <group>INFO</group>
        <nbTotal>16</nbTotal>
        <nbLevel5>0</nbLevel5>
        <nbLevel4>1</nbLevel4>
        <nbLevel3>11</nbLevel3>
        <nbLevel2>4</nbLevel2>
        <nbLevel1>0</nbLevel1>
    </GroupStat>
</list>undefined</byGroup>undefined<byOwasp>
<count>2</count>
<list>
    <OwaspStat>
        <owasp>OWASP-A2</owasp>
        <nbTotal>2</nbTotal>
        <nbLevel5>0</nbLevel5>
        <nbLevel4>0</nbLevel4>
        <nbLevel3>1</nbLevel3>
        <nbLevel2>1</nbLevel2>
        <nbLevel1>0</nbLevel1>
    </OwaspStat>
    <OwaspStat>
        <owasp>OWASP-A5</owasp>
        <nbTotal>14</nbTotal>
        <nbLevel5>0</nbLevel5>
        <nbLevel4>1</nbLevel4>
        <nbLevel3>10</nbLevel3>
        <nbLevel2>3</nbLevel2>
        <nbLevel1>0</nbLevel1>
    </OwaspStat>
</list>undefined</byOwasp>
.........
.........
```

```
 undefined<group>INFO</group>undefined</WasScanVuln>undefined<WasScanV
uln>undefined<qid>150124</qid>undefined<severity>3</severity>undefined
<potential>false</potential>undefined<title>
<![CDATA[Clickjacking - Framable
Page]]>undefined</title>undefined<uri>
<![CDATA[http://funkytown.vuln.qa.qualys.com/cassium/sql/]]>undefined<
/uri>undefined<instances>
<count>1</count>
<list>
    <WasScanVulnInstance>
        <authenticated>false</authenticated>
        <payloads>
            <count>1</count>
            <list>
                <WasScanVulnPayload>
                    <payload>
                        <![CDATA[N/A]]>
                    </payload>
                    <result base64="true">
                        <![CDATA[VGhlIFVSSSB3YXMgZnJhbWVkLgo=]]>
                    </result>
                </WasScanVulnPayload>
            </list>
        </payloads>
    </WasScanVulnInstance>
</list>undefined</instances>
.......
.......
.......

undefined<![CDATA[TGludXhfMi40LTIuNl8vX0VtYmVkZGVkX0RldmljZV8vX0Y1X05l
dHdvcmtzX0Jp
Zy1JUF8vX1JlZF9IYXRfRW50ZXJwcmlzZV9MaW51eF9TZXJ2ZXJfcmVsZWFzZV81LjJfKF
Rpa
2FuZ2EpIFRDUC9JUF9GaW5nZXJwcmludCBNMTE0MTo3MzIwOjo4MAo=]]>undefined</d
ata>undefined<sslData>undefined<protocol>tcp</protocol>undefined<ip>10
.11.68.108</ip>undefined<result>
<![CDATA[Linux_2.4-
2.6_/_Embedded_Device_/_F5_Networks_BigIP_/_Red_Hat_Enterprise_Linux_S
erver_release_5.2_(Tikanga)
TCP/IP_Fingerprint M1141:7320::80
]]>undefined</result>undefined<sslDataInfoList>
<list>
    <SSLDataInfo>
        <sslDataCipherList/>
        <sslDataKexList/>
        <sslDataPropList/>
    </SSLDataInfo>
</list>undefined</sslDataInfoList>undefined</sslData>undefined</WasSca
nIg>undefined</list>undefined</igs>undefined<sendMail>true</sendMail>u
ndefined<enableWAFAuth>false</enableWAFAuth>undefined</WasScan>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscan.xsd

Cancel Scan

/qps/rest/3.0/cancel/was/wasscan/<id>

[POST]

Cancel an unfinished scan on a web application which is in the user’s scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "Cancel WAS Scan".

Input Parameters

The element “id” (integer) is required, where “id” identifies the scan.

Click here for available operators

Sample - Cancel unfinished scan

Cancel the unfinished scan that has the ID 168.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/cancel/was/wasscan/168"
```

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/3.
0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>168</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Sample - Cancel unfinished scan with scan results

Use parameter <cancelWithResults> to cancel the scan and still retain results. You can use the scan ID and generate a report to view the results.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/cancel/was/wasscan/6620298"
```

Request POST data

```
<ServiceRequest>
  <data>
        <WasScan>
            <cancelWithResults>true</cancelWithResults>
        </WasScan>
    </data>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
     <responseCode>SUCCESS</responseCode>
     <data>
        <WasScan>
            <id>6620298</id>
        </WasScan>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscan.xsd

Delete Scan

/qps/rest/3.0/delete/was/wasscan/<id>

/qps/rest/3.0/delete/was/wasscan/<filters>

[POST]

Delete an existing scan on a web application which is in the user's scope. You can delete any scan in your account that is not running.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access" and "Delete WAS scan" permission. The scan to be deleted must be within the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. Click here for descriptions of <WebApp> elements

Click here for available operators

Parameter	Description
id	(integer)The scan ID.
name	(text) The scan name.
webApp.name	(text) The name of the web application being scanned.
webApp.id	(integer) The ID of the web application being scanned.
reference	(text) Scan Reference ID.
launchedDate	(date) The date and time when the scan was launched in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

type	(keyword) The scan type: VULNERABILITY or DISCOVERY.
mode	(keyword) The mode of the scan: ONDEMAND, SCHEDULED or API.
status	(keyword) The status of the scan: SUBMITTED, RUNNING, FINISHED, ERROR, CANCELED, PROCESSING.
authStatus	(Keyword) Indicates the status of the authentication record: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL.
resultsStatus	(keyword) The status of the scan: NOT_USED, TO_BE_PROCESSED, NO_HOST_ALIVE, NO_WEB_SERVICE, SERVICE_ERROR, TIME_LIMIT_REACHED, SCAN_INTERNAL_ERROR, SCAN_RESULTS_INVALID, SUCCESSFUL, PROCESSING, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, SUBMITTED, RUNNING, FINISHED, CANCELED, CANCELING, ERROR, DELETED, CANCELED_WITH_RESULTS.

Sample - Delete a specified scan

Let us delete the scan with the ID 12405.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/wasscan/12405"
```

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/3.
0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScan>
      <id>12405</id>
    </WasScan>
  </data>
</ServiceResponse>
```

Sample - Delete scans with criteria

Let us delete scans with a name that contains the string “VULN”.

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
         <Criteria field="name" operator="CONTAINS">VULN</Criteria>
    </filters>
</ServiceRequest>
```

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/3.
0/was/wasscan.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <data>
    <WasScan>
      <id>12874</id>
    </WasScan>
    <WasScan>
      <id>13093</id>
    </WasScan>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscan.xsd

WasScan Reference

The <WasScan> element includes sub elements used to define a web application scan. A reference of these elements is provided below. An asterisk * indicates a complex element.

Parameter	Description
id	(integer) The scan ID. This element is assigned by the service and is required for a certain type of request (details, status, results or cancel).
name	(text) The user-defined scan name (maximum 256 characters).

target* (for single web application)	(text) The target of the scan. The target includes the web application and authentication records, if any.
	<scannerAppliance> - type (keyword) is set to INTERNAL for a scanner appliance, or EXTERNAL for external scanners or scannerTags for assigning multiple scanner appliances grouped by asset tag. If the type is INTERNAL, friendlyName (text) is the user-defined appliance name.
	</webAppAuthRecord> - Specify <id> set to an auth record ID, or <isDefault> set to true (to use the default auth record for the target web app).
	`Example: target.webApp is required`
	`<target>`
	`    <webApp>`
	`        <id>323126</id>`
	`    </webApp>`
	`<webAppAuthRecord>`
	`        <id>1054</id>`
	`    </webAppAuthRecord>`
	`    <scannerAppliance>`
	`        <type>Internal</type>`
	`        <friendlyName>dp_scanner</friendlyName>`
	`    </scannerAppliance>`

```
    <cancelOption>DEFAULT</cancelOption>
</target>
```

target* (for multiple web application)	<cancelOption> set to DEFAULT - Forces the use of the target web app’s cancelScans option if set, else fall back to the one passed in to the API while launching the scan. <cancelOption> set to SPECIFIC - Always use the cancel scan option passed while launching the scan. <target.authRecordOption> set to SPECIFIC -Always use the authRecord passed while launching the scan. <target.authRecordOption> set to DEFAULT-Forces the use of the authRecord, if set, else fall back to the one passed in to the API while launching the scan. <target.profileOption> set to SPECIFIC-Always use the optionProfile passed while launching the scan. <target.profileOption> set to DEFAULT-Forces the use of the optionProfile if set, else fall back to the one passed in to the API while launching the scan. <target.scannerOption> set to SPECIFIC-Always use the scanner passed while launching the scan. <target.scannerOption> set to DEFAULTForces the use of the scanner if set, else fall back to the one passed in to the API while launching the scan. <target.randomizeScan> (Boolean) - Set to true to scan the selected web applications in random order. Set to false to scan the selected web application in sequential order. target.tags (For MultiScan)-- ---target.tags.included.option(ALL/ANY) is required, ---target.tags.included.tagList is required, only <set> is allowed for target.tags.included.tagList.

--- target.tags.included.tagList.set.Tag.id is required and should be valid

---Only target.tags.exclusive is not allowed, it must be with target.tags.inclusive

---If target.tags.excluded is present, all the above rules are applicable to it

Example: Either target.webApps or target.tags is required and these are mutually exclusive.

```
target.webApps (For MultiScan)-
Only <set> is allowed for target.webApps
 <webApps>
   <set>
      <WebApp>
         <id>4330527</id>
      </WebApp>
      <WebApp>
         <id>4330327</id>
      </WebApp>
   </set>
  </webApps>
target.tags (For MultiScan)-
<tags>
     <included>
       <option>ALL</option>
         <tagList>
           <set>
             <Tag><id>12017424</id></Tag>
             <Tag><id>12017228</id></Tag>
           </set>
         </tagList>
         </included>
         <excluded>
            <option>ANY</option>
                <tagList>
                   <set>
                     <Tag><id>12017228</id></Tag>
                 </set>
                </tagList>
         </excluded>
   </tags>
```

type	(keyword) The scan type: VULNERABILITY or DISCOVERY.
sendMail	(boolean) Set to false to disable scan complete email notifications. `Example:<sendMail>false</sendMail>`
sendOneMail	(boolean) Set to false to disable scan complete email notifications. `Example:<sendMail>false</sendMail>`

profile.id	(integer) The name of the option profile that includes scan settings. The service provides the profile “Initial WAS Options” and we recommend this to get started.
	`Example:`
	`<profile>`
	`    <name>Initial WAS Options</name>`
	`</profile>`

proxy.id	(integer) The name of the option profile that includes scan settings. The service provides the profile “Initial WAS Options” and we recommend this to get started.
	`Example:`
	`<profile>`
	`    <name>Initial WAS Options</name>`
	`</profile>`

dnsOverride.id	(integer) The DNS override record for scanning the target web application.
	`Example:`
	`<dnsOverride>`
	`   <id>67890</id>`
	`</dnsOverride>`

Scanner Appliance	(integer)The IP address of the external scanner appliance, when an external scanner is used.

mode

(keyword) The mode of the scan: ONDEMAND, SCHEDULED or API.

launchedDate

(date) The date and time when the scan was launched in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

launchedBy*

The user who launched the scan. User properties include user ID, user login, first and last name.

```
Example:
<launchedBy>
  <id>123056</id>
  <username>username</username>
  <firstName><![CDATA[John]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</launchedBy>
```

status

(keyword) The status of the scan: SUBMITTED, RUNNING, FINISHED, ERROR, CANCELED, PROCESSING.

endScanDate

(date) The date and time when the scan ended in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

summary

The scan summary. <crawlTime> is the length of time used to crawl the web application. <testDuration> is the length of time used to perform analysis. <nbRequests> is the number of requests sent during the scan. <authStatus> is the authentication status (NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL)

Example:

```
<summary>
  <crawlTime>22.0</crawlTime>
  <testDuration>112.0</testTime>
  <linksCrawled>17</linksCrawled>
  <nbRequests>3814</nbRequests>
<os>Windows XP SP2</os>
<resultsStatus>RESULTS_PROCESSED_SUCCESSFULLY</resul
tsStatus>
  <authStatus>NO_AUTH</authStatus>
</summary>
```

vulns	The list of detected vulnerabilities. Each <WasScanVuln> element identifies a particular vulnerability QID and the URI where detected, each <WasScanVulnInstance> element identifies a vulnerability instance, and each <WasScanVulnInstancePayload> element identifies associated payloads.
igs	The detected information gathered. Each <WasScanIg> element identifies a particular information gathered QID.
sensitiveContents	The detected sensitive content. Each <WasScanSensitiveContent> element identifies a particular sensitive content QID and the URI where detected, each <instances> element identifies a sensitive content instance, and each <WasScanSensitiveContentInstancePayLoad> element identifies associated payloads.
stats	The statistics gathered by the scan: the total number of vulnerabilities, the number of vulnerabilities by severity level, information gathered by severity level and the number of vulnerabilities by group, OWASP and WASC.

cancelWithResults	(boolean) A flag to indicate if the scan to be canceled should retain partial scan results or not. The parameter is supported for single scan, only child scan (but not parent scan).
	We recommend you to use this parameter only after 20 minutes of scan goes into Running status.
	Example:
	`<WasScan>`
	`  <cancelWithResults>true</cancelWithResults>`
	`</WasScan>`

WAS Scan Results Reference

You have the option to retrieve web application scan results in legacy format (WAS v2 and earlier), using the webapp_scan.dtd (see Retrieve the results of a scan). You can download this DTD by going to https://qualysapi.qualys.com/webapp_scan.dtd (where qualysapi is the API server URL where your account is located).

WAS scan results DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS WEB APPLICATION SCAN DTD -->
<!ELEMENT WEB_APPLICATION_SCAN (ERROR | (HEADER, SUMMARY,
                                RESULTS))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>
<!-- GENERIC HEADER -->
<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO,
                  USER_INFO)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>
<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY,
                        ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>
<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>
<!-- SUMMARY -->
<!ELEMENT SUMMARY (SCAN_SUMMARY, VULN_SUMMARY?,
                   SENSITIVE_CONTENT_SUMMARY)>
<!ELEMENT SCAN_SUMMARY (SCAN_INFO*)>
<!ELEMENT SCAN_INFO (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!ELEMENT VULN_SUMMARY (VULN_GROUP*)>
<!ELEMENT VULN_GROUP (TITLE, SEVERITY_5, SEVERITY_4, SEVERITY_3,
                      SEVERITY_2, SEVERITY_1, TOTAL)>
<!ELEMENT SEVERITY_1 (#PCDATA)>
<!ELEMENT SEVERITY_2 (#PCDATA)>
<!ELEMENT SEVERITY_3 (#PCDATA)>
<!ELEMENT SEVERITY_4 (#PCDATA)>
```

```
<!ELEMENT SEVERITY_5 (#PCDATA)>
<!ELEMENT TOTAL (#PCDATA)>
<!ELEMENT SENSITIVE_CONTENT_SUMMARY (SENSITIVE_CONTENT_GROUP*)>
<!ELEMENT SENSITIVE_CONTENT_GROUP (TITLE, TOTAL)>
<!-- RESULTS -->
<!ELEMENT RESULTS (VULN_LIST?, SENSITIVE_CONTENT_LIST?,
                   INFO_LIST?)>
<!ELEMENT VULN_LIST (VULN*)>
<!ELEMENT VULN (GROUP, QID, TITLE, VULN_INSTANCES)>
<!ELEMENT VULN_INSTANCES (VULN_INSTANCE*)>
<!ELEMENT VULN_INSTANCE (HOST, PORT, URI, AUTHENTICATED?,
FORM_ENTRY_POINT?, PARAMS, FINDINGS)>
<!ELEMENT AUTHENTICATED (#PCDATA)>
<!ELEMENT FORM_ENTRY_POINT (#PCDATA)>
<!ELEMENT SENSITIVE_CONTENT_LIST (SENSITIVE_CONTENT*)>
<!ELEMENT SENSITIVE_CONTENT (GROUP, QID, TITLE,
                             SENSITIVE_CONTENT_INSTANCES)>
<!ELEMENT SENSITIVE_CONTENT_INSTANCES (SENSITIVE_CONTENT_INSTANCE*)>
<!ELEMENT SENSITIVE_CONTENT_INSTANCE (HOST, PORT, URI, CONTENT?,
                                      FINDINGS)>
<!ELEMENT INFO_LIST (INFO*)>
<!ELEMENT INFO (QID, TITLE, RESULT)>
<!ELEMENT GROUP (#PCDATA)>
<!ELEMENT QID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT HOST (#PCDATA)>
<!ELEMENT PORT (#PCDATA)>
<!ELEMENT URI (#PCDATA)>
<!ELEMENT CONTENT (#PCDATA)>
<!ELEMENT PARAMS (#PCDATA)>
<!ELEMENT FINDINGS (FINDING*)>
<!ELEMENT FINDING (PAYLOAD?, RESULT)>
<!ELEMENT PAYLOAD (#PCDATA)>
<!ELEMENT RESULT (#PCDATA)>
<!ATTLIST RESULT base64 (true|false) "false">
```

Schedules

Schedule Count

/qps/rest/3.0/count/was/wasscanschedule

[GET] [POST]

Returns the total number of schedules in the user's account. Input elements are optional and are used to filter the number of schedules included in the count.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes scan targets in the user's scope.

Input Parameters

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. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements.

Click here for available operators

Parameter	Description
id	(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.
name	(text) The user-defined schedule name (maximum 256 characters).
owner.id	(integer) ID associated with the owner who created the schedule.

createdDate	(date) The date when the schedule was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the schedule was created in WAS, in UTC date/time format.
type	(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.
webApp.name	(text) The name of the web application being scanned.
webApp.id	(integer) The ID of the web application being scanned.
webApp.tags (with operator="NONE")	Tags associated with the web application being scanned.
webApp.tags.id	(integer) ID of the tag applied to the web application being scanned.
invalid	(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.
active	(boolean) Indicates whether the schedule is active or not. True indicates active schedule.

Sample - Get count of schedules in user's account

Return the number (count) of all schedules in the user's scope.

API request

```
curl -u "USERNAME:PASSWORD"
https://qualysapi.qualys.com/qps/rest/3.0/count/was/wasscanschedule"
```

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/3.
0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>15</count>
</ServiceResponse>
```

Sample - Get count of schedules with a criteria

Return the number (count) of schedules for discovery scan type.

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="type" operator="EQUALS">DISCOVERY</Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
</ServiceResponse>
```

Sample - Get count of schedules for web applications without tags

Return the number (count) of schedules for web application that are not tagged..

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags" operator="NONE"></Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```

Sample - Get count of schedules for web applications with tags

Return the number (count) of schedules for web applications that are tagged..

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
        <Criteria field="webApp.tags.id"
operator="EQUALS">1516928</Criteria>
```

```
        <Criteria field="webApp.tags.id"
operator="EQUALS">1234567</Criteria>
    </filters>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscanschedule.xsd

Search Schedule

/qps/rest/3.0/search/was/wasscanschedule

[POST]

Returns a list of scheduled scans on web applications which are in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes scan targets in the user's scope.

Input Parameters

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. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements.

Click here for available operators

Parameter	Description
id	(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.
name	(text) The user-defined schedule name (maximum 256 characters).
owner.id	(integer) ID associated with the owner who created the schedule.
createdDate	(date) The date when the schedule was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the schedule was created in WAS, in UTC date/time format.

active	(boolean) Indicates whether the schedule is active or not. True indicates active schedule.
type	(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.
webApp.name	(text) The name of the web application being scanned.
webApp.id	(integer) The ID of the web application being scanned.
webApp.tags (with operator="NONE")	Tags associated with the web application being scanned.
webApp.tags.id	(integer) ID of the tag applied to the web application being scanned.
invalid	(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.
lastScan (with operation="NONE")	(boolean) Indicates if the last scan was performed or not. True indicates that the last scan was performed.
lastScan.launchedDate	(date) Date when the last scan was launched on the web application, in UTC date/time format.
lastScan.status	(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR, CANCELED)
multi	(boolean) Indicates if the scheduled scan is single scan or multiple scan.

Sample - List of schedules never launched

Let us view a list of all schedules that are in the user’s scope but were not launched.

API request

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

Request POST data

```
<ServiceRequest>
      <filters>
          <Criteria field="lastScan" operator="NONE"></Criteria>
      </filters>
</ServiceRequest>
```

XML 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/3.0
/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <WasScanSchedule>
      <id>171425669</id>
      <name><![CDATA[Web Application Vulnerability Scan - 2017-Aug-
19]]></name>
      <owner>
        <id>8792415669</id>
      </owner>
      <active>false</active>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
          <id>1296335669</id>
          <name><![CDATA[My Web Application]]></name>
          <url><![CDATA[http://10.10.1.100]]></url>
        </webApp>
        <webAppAuthRecord>
          <id>175535669</id>
          <name><![CDATA[AR1]]></name>
        </webAppAuthRecord>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
      </target>
      <profile>
        <id>716315669</id>
        <name><![CDATA[Copy of Initial WAS Options]]></name>
      </profile>
      <scheduling>
        <startDate>2017-08-19T12:30:00Z</startDate>
        <timeZone>
          <code>America/Dawson</code>
          <offset>-07:00</offset>
        </timeZone>
        <occurrenceType>ONCE</occurrenceType>
      </scheduling>
      <createdDate>2017-08-19T19:30:49Z</createdDate>
      <updatedDate>2017-08-19T19:30:50Z</updatedDate>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Sample - List launched schedules

Let us view a list of all schedules that are in the user's scope and were launched.

API request

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

Request POST data

```
<ServiceRequest>
        <filters>
        <Criteria field="lastScan.status"
operator="IN">FINISHED,ERROR</Criteria>
```

```
        <Criteria field="lastScan.launchedDate"
operator="LESSER">2017-08-19</Criteria>
        </filters>
</ServiceRequest>
```

XML response

```
…
    </WasScanSchedule>
    <WasScanSchedule>
      <id>97354000</id>
      <name><![CDATA[Schedule Notification]]></name>
      <owner>
        <id>334527</id>
      </owner>
      <active>false</active>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
          <id>1061764000</id>
          <name><![CDATA[My Web App]]></name>
          <url><![CDATA[http://10.10.26.238]]></url>
        </webApp>
        <webAppAuthRecord>
          <id>8753</id>
          <name><![CDATA[Auth Record 1]]></name>
        </webAppAuthRecord>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
      </target>
      <profile>
        <id>55784</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <scheduling>
        <startDate>2017-05-06T18:22:00Z</startDate>
        <timeZone>
          <code>America/Dawson</code>
          <offset>-07:00</offset>
        </timeZone>
        <occurrenceType>DAILY</occurrenceType>
        <occurrence>
          <dailyOccurrence>
            <everyNDays>1</everyNDays>
          </dailyOccurrence>
        </occurrence>
      </scheduling>
      <lastScan>
        <id>14929668885</id>
        <launchedDate>2017-05-12T01:22:02Z</launchedDate>
        <status>FINISHED</status>
      </lastScan>
      <createdDate>2017-05-06T23:17:23Z</createdDate>
      <updatedDate>2017-05-13T01:22:02Z</updatedDate>
    </WasScanSchedule>
…
```

Sample - List schedules no criteria

Let us view a list of all schedules that are in the user’s scope and were launched.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/wasscanschedule"
< file.xml
Note: “file.xml” contains the request POST data. Specify an empty
file, since no search criteria is being specified.
```

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/3.
0/was/wasscanschedule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <WasScanSchedule>
            <id>649146</id>
            <name>
                <![CDATA[Web Application Vulnerability Scan - 2018-10-
08]]>
            </name>
            <owner>
                <id>412791</id>
            </owner>
            <active>true</active>
            <multi>false</multi>
            <type>VULNERABILITY</type>
            <target>
                <webApp>
                    <id>8077389</id>
                    <name>
                        <![CDATA[SampleWebApp_1538665472012 ]]>
                    </name>
                    <url>
                        <![CDATA[http://funkytown.vuln.qa.example.com:
80/cassium/xss/]]>
                    </url>
                </webApp>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <cancelOption>SPECIFIC</cancelOption>
            </target>
            <profile>
                <id>1162483</id>
                <name>
                    <![CDATA[Option Profile]]>
                </name>
            </profile>
            <scheduling>
                <startDate>2018-10-08T16:41:00Z</startDate>
                <timeZone>
                    <code>Asia/Colombo</code>
                    <offset>+05:30</offset>
                </timeZone>
                <occurrenceType>ONCE</occurrenceType>
            </scheduling>
            <nextLaunchDate>2018-10-09T11:11:00Z</nextLaunchDate>
            <createdDate>2018-10-08T11:12:28Z</createdDate>
            <updatedDate>2018-10-08T11:12:29Z</updatedDate>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```

Sample - List active schedules

Let us view a list of all schedules that are in the user's scope and were launched.

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
        <Criteria field="active" operator="EQUALS">true</Criteria>
        <Criteria field="type"
operator="EQUALS">VULNERABILITY</Criteria>
    </filters>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <WasScanSchedule>
            <id>649146</id>
            <name>
                <![CDATA[Web Application Vulnerability Scan - 2018-10-
08]]>
            </name>
            <owner>
                <id>412791</id>
            </owner>
            <active>true</active>
            <multi>false</multi>
            <type>VULNERABILITY</type>
            <target>
                <webApp>
```

```
                    <id>8077389</id>
                    <name>
                        <![CDATA[SampleWebApp_1538665472012 ]]>
                    </name>
                    <url>
                        <![CDATA[http://funkytown.vuln.qa.example.com:
80/cassium/xss/]]>
                    </url>
                </webApp>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <cancelOption>SPECIFIC</cancelOption>
            </target>
            <profile>
                <id>1162483</id>
                <name>
                    <![CDATA[Option Profile]]>
                </name>
            </profile>
            <scheduling>
                <startDate>2018-10-08T16:41:00Z</startDate>
                <timeZone>
                    <code>Asia/Colombo</code>
                    <offset>+05:30</offset>
                </timeZone>
                <occurrenceType>ONCE</occurrenceType>
            </scheduling>
            <nextLaunchDate>2018-10-09T11:11:00Z</nextLaunchDate>
            <createdDate>2018-10-08T11:12:28Z</createdDate>
            <updatedDate>2018-10-08T11:12:29Z</updatedDate>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/webappauthrecord.xsd

Get Schedule Details

/qps/rest/3.0/get/was/wasscanschedule/<id>

[GET]

View details for a scheduled scan on a web application which is in the user's scope. Want to find a schedule ID to use as input? See Search schedules.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes schedules in the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies a schedule.

Click here for available operators

Sample - View schedule details

Let us view details for schedule with ID 714393.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscanschedule/714
393"
```

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/3.
0/was/wasscanschedule.xsd">
<responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScanSchedule>
            <id>714393</id>
            <name>
                <![CDATA[Web schedVulnerability Scan - 2017-06-30]]>
            </name>
            <owner>
                <id>2473353</id>
                <username>username</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Smith]]></lastName>
            </owner>
            <active>false</active>
            <multi>true</multi>
            <type>VULNERABILITY</type>
            <target>
                <tags>
                    <included>
                        <option>ALL</option>
                        <tagList>
                            <list>
                                <Tag>
                                    <id>12075819</id>
                                    <name>
                                        <![CDATA[New_tag]]>
                                    </name>
                                </Tag>
                                <Tag>
                                    <id>2685657</id>
                                    <name>
                                        <![CDATA[Business Units]]>
                                    </name>
                                </Tag>
                            </list>
                        </tagList>
                    </included>
                </tags>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <cancelOption>DEFAULT</cancelOption>
                <authRecordOption>DEFAULT</authRecordOption>
                <profileOption>DEFAULT</profileOption>
                <scannerOption>DEFAULT</scannerOption>
                <randomizeScan>false</randomizeScan>
                <useDnsOverride>false</useDnsOverride>
            </target>
            <profile>
                <id>598333</id>
                <name>
```

```
                    <![CDATA[Initial WAS Options]]>
                </name>
            </profile>
            <scheduling>
                <startDate>2017-06-30T11:26:00Z</startDate>
                <timeZone>
                    <code>Asia/Colombo</code>
                    <offset>+05:30</offset>
                </timeZone>
                <occurrenceType>ONCE</occurrenceType>
            </scheduling>
            <notification>
                <active>false</active>
                <reschedule>false</reschedule>
                <delay>
                    <nb>1</nb>
                    <scale>DAY</scale>
                </delay>
                <message>
               <![CDATA[A Qualys scan is scheduled to start soon.]]>
                </message>
            </notification>
            <launchedCount>0</launchedCount>
            <createdDate>2017-06-30T05:57:12Z</createdDate>
            <createdBy>
                <id>2473353</id>
                <username>username</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Smith]]></lastName>
            </createdBy>
            <updatedDate>2017-07-01T05:56:02Z</updatedDate>
            <updatedBy>
                <id>2473353</id>
                 <username>username</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Smith]]></lastName>
            </updatedBy>
            <sendMail>true</sendMail>
            <sendOneMail>true</sendOneMail>
            <enableWAFAuth>false</enableWAFAuth>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```

Sample - View schedule details (progressive scan)

The progressiveScanning element will be included in the call response, if Progressive Scanning is enabled for the subscription.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/wasscanschedule/818
3"
```

XML response

```
<?xml version="1.0" encoding=<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.
0/was/wasscanschedule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScanSchedule>
            <id>8183</id>
            <name>
                <![CDATA[WASUI-3772 #3]]>
            </name>
...
            <progressiveScanning>ENABLED</progressiveScanning>
...
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscanschedule.xsd

Create a Schedule (single web application)

/qps/rest/3.0/create/was/wasscanschedule

[POST]

Create a scheduled scan on a web application which is in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access" and "Create WAS Schedule" permission. The output includes schedules in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements

Click here for available operators

Parameter	Description
name	(text) Name of the schedule.
target.webApp.id[1]	(integer) The web applications to be scanned.
type	(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.

profile.id[2]	(integer) The name of the option profile that includes scan settings. The service provides the profile "Initial WAS Options" and we recommend this to get started.
	`Example:`
	`<profile>`
	`    <name>Initial WAS Options</name>`
	`</profile>`

startDate	(date) The date when the schedule starts in UTC date/time format.
timeZone	(text) The timezone in which the scan is scheduled in UTC date/time format.
occurrenceType	(keyword) The frequency of the scheduled scan : ONCE, DAILY, WEEKLY or MONTHLY.
notification	(boolean)A flag indicating whether email notification is enabled for scheduled scan.
reschedule	(boolean) Set this flag to reschedule the scan.
target.scannerAppliance.type	(keyword) The type of scanner appliance used for the scan: EXTERNAL or INTERNAL or scannerTags.
target.scannerAppliance.friendlyName	(text) Name of the scanner appliance used for the scan.
target.scannerTags.set.Tag.id	(integer) The scanner associated with the tag (identified by the specified tag ID) is picked for the scan.
target.webAppAuthRecord.id or target.webAppAuthRecord.isDefault	Decides the authentication record to be used for the scan. target.webAppAuthRecord.id (integer): Specify the web application's authentication record ID to use the specific authentication record. target.webAppAuthRecord.isDefault (boolean): Set to true to use the default web application's authentication record for the scan.

options	(keyword: ANY, ALL) Decides which web applications should be excluded from the scan. ALL : Only the web applications associated with all the specified tags are excluded from the scan. ANY : Only the web applications associated with any of the specified tags are excluded from the scan.

proxy.id	(integer) The proxy for scanning the target web application.
	`Example:`
	`<proxy>`
	`   <id>12345</id>`
	`</proxy>`

dnsOverride.id	(integer) The DNS override record for scanning the target web application.
	`Example:`
	`<dnsOverride>`
	`   <id>67890</id>`
	`</dnsOverride>`

cancelOption	(keyword: DEFAULT, SPECIFIC) set to DEFAULT - Forces the use of the target web app’s cancelScans option if set, else fall back to the one passed in to the API while launching the scan. set to SPECIFIC - Always use the cancel scan option passed while launching the scan.
sendMail	(boolean) Set to false to disable scan complete email notifications.

	`Example:<sendMail>false</sendMail>`

sendMailFromAddressOption	Identifies the sender of the scan complete notifications. The valid values are: QUALYS_SUPPORT and OWNER. OWNER means the user whose account is used to create the schedule.
	`Example:<sendMailFromAddressOption>QUALYS_ SUPPORT</sendMailFromAddressOption>`
	`Example:<sendMailFromAddressOption>OWNER </sendMailFromAddressOption>`
	To set this parameter, the sendMail parameter must be set to true. If the sendMail parameter is true, then sendMailFromAddressOption is by default set to QUALYS_SUPPORT. You can change the value of the parameter to OWNER.

1 The element target must have at least tags or web applications specified.

2 The element profile (text) is required unless the target has a default option profile.

Sample - Create a new weekly schedule

Let us create a new web application called “My Web Application” that has the starting URL “http://mywebapp.com”. The default web application settings are assigned automatically.

API request

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

Request POST data

```
<ServiceRequest>
 <data>
   <WasScanSchedule>
           <name><![CDATA[Create Schedule from API3 - using 
Reschedule]]></name>
     <type>VULNERABILITY</type>
     <active>false</active>
     <scheduling>
       <cancelAfterNHours>8</cancelAfterNHours>
       <startDate>2017-09-06T09:50:11Z</startDate>
       <timeZone>
         <code>America/Vancouver</code>
        <offset>-07:00</offset>
       </timeZone>
       <occurrenceType>WEEKLY</occurrenceType>
       <occurrence>
          <weeklyOccurrence>
            <everyNWeeks>2</everyNWeeks>
            <occurrenceCount>20</occurrenceCount>
            <onDays>
              <WeekDay>SATURDAY</WeekDay>
              <WeekDay>SUNDAY</WeekDay>
            </onDays>
          </weeklyOccurrence>
       </occurrence>
     </scheduling>
    <notification>
        <active>true</active>
        <reschedule>true</reschedule>
        <delay>
          <nb>1</nb>
          <scale>DAY</scale>
        </delay>
        <message><![CDATA[A Qualys scan is scheduled to start 
soon.]]></message>
      </notification>
     <target>
       <webApp>
              <id>1296335669</id>
</webApp>
        <webAppAuthRecord>
              <id>175535669</id>
</webAppAuthRecord>
     </target>
     <profile>
```

```
<id>712265669</id>
</profile>
   </WasScanSchedule>
 </data>
</ServiceRequest>
```

XML 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/3.0
/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>203285669</id>
      <name><![CDATA[Create Schedule from API3 - using
Reschedule]]></name>
      <owner>
        <id>8792415669</id>
        <username>joe_user</username>
        <firstName><![CDATA[Customer_2.6_1]]></firstName>
        <lastName><![CDATA[pocm]]></lastName>
      </owner>
      <active>false</active>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
          <id>1296335669</id>
          <name><![CDATA[My Web Application]]></name>
          <url><![CDATA[http://10.10.26.238]]></url>
        </webApp>
        <webAppAuthRecord>
          <id>175535669</id>
          <name><![CDATA[AR1]]></name>
        </webAppAuthRecord>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
      </target>
      <profile>
        <id>712265669</id>
        <name><![CDATA[Initial WAS Options]]></name>
      </profile>
      <scheduling>
        <startDate>2017-09-06T09:50:00Z</startDate>
        <timeZone>
          <code>America/Vancouver</code>
          <offset>-07:00</offset>
        </timeZone>
        <occurrenceType>ONCE</occurrenceType>
        <cancelAfterNHours>8</cancelAfterNHours>
      </scheduling>
      <notification>
        <active>true</active>
        <reschedule>true</reschedule>
        <delay>
          <nb>1</nb>
          <scale>DAY</scale>
        </delay>
        <message><![CDATA[A Qualys scan is scheduled to start 
soon.]]></message>
      </notification>
      <launchedCount>0</launchedCount>
      <createdDate>2017-08-27T22:30:59Z</createdDate>
      <createdBy>
        <id>8792415669</id>
        <username>john_doe</username>
        <firstName><![CDATA[Customer_2.6_1]]></firstName>
        <lastName><![CDATA[doe]]></lastName>
      </createdBy>
      <updatedDate>2017-08-27T22:31:00Z</updatedDate>
      <updatedBy>
        <id>8792415669</id>
        <username>user_john</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
      <sendMail>true</sendMail>
      <sendOneMail>true</sendOneMail>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Sample - Create a new schedule - cancel scan option

Create a new vulnerability scan schedule on web app ID 2376281 and set the cancel scan option to SPECIFIC. Scans launched from this schedule will always use the cancel scan option passed with the schedule settings and will override the target web app’s cancel scan setting, if set.

API request

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

Request POST data

```
<ServiceRequest>
 <data>
   <WasScanSchedule>
     <name><![CDATA[My Scan Schedule]]></name>
     <type>VULNERABILITY</type>
     <scheduling>
        <cancelAfterNHours>7</cancelAfterNHours>
       <startDate>2017-09-30T13:11:00Z</startDate>
       <timeZone>
         <code>America/Dawson</code>
       </timeZone>
       <occurrenceType>ONCE</occurrenceType>
     </scheduling>
     <target>
       <webApp>
           <id>2376281</id>
       </webApp>
       <scannerAppliance>
         <type>EXTERNAL</type>
       </scannerAppliance>
       <cancelOption>SPECIFIC</cancelOption>
     </target>
     <profile>
        <id>332147</id>
     </profile>
   </WasScanSchedule>
 </data>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>325624</id>
      <name><![CDATA[My Scan Schedule]]></name>
      <owner>
        <id>2086786</id>
        <username>user_john</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Doe]]></lastName>
      </owner>
      <active>true</active>
      <type>VULNERABILITY</type>
      <target>
        <webApp>
          <id>2376281</id>
          <name><![CDATA[My Web App]]></name>
          <url><![CDATA[http://10.10.26.238]]></url>
        </webApp>
        <scannerAppliance>
          <type>EXTERNAL</type>
        </scannerAppliance>
        <cancelOption>SPECIFIC</cancelOption>
      </target>
      <progressiveScanning>DEFAULT</progressiveScanning>
      <profile>
        <id>332147</id>
        <name><![CDATA[10 links]]></name>
      </profile>
      <scheduling>
        <startDate>2017-09-30T13:11:00Z</startDate>
        <timeZone>
          <code>America/Dawson</code>
          <offset>-07:00</offset>
        </timeZone>
        <occurrenceType>ONCE</occurrenceType>
        <cancelAfterNHours>7</cancelAfterNHours>
      </scheduling>
      <notification>
        <active>false</active>
      </notification>
      <nextLaunchDate>2017-09-30T20:11:00Z</nextLaunchDate>
      <launchedCount>0</launchedCount>
      <createdDate>2017-06-26T20:54:30Z</createdDate>
      <createdBy>
        <id>2086786</id>
         <username>user_john</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Doe]]></lastName>
      </createdBy>
      <updatedDate>2017-06-26T20:54:30Z</updatedDate>
      <updatedBy>
        <id>2086786</id>
        <username>user_john</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Doe]]></lastName>
      </updatedBy>
      <sendMail>true</sendMail>
      <sendOneMail>false</sendOneMail>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Sample - Create a new schedule - assign multiple scanners

Let us schedule a discovery scan on the web application and assign the pool of scanners using the asset tag ID.

API request

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

Request POST data

```
<ServiceRequest>
   <data>
     <WasScanSchedule>
      <name><![CDATA[Scheduled Scan With Pool of Internal Scanners]>
      </name>
      <type>VULNERABILITY</type>
    <active>false</active>
    <scheduling>
```

```
      <cancelAfterNHours>10</cancelAfterNHours>
      <startDate>2017-01-10T13:55:35Z</startDate>
      <timeZone>
        <code>Europe/Istanbul</code>
        <offset>+02:00</offset>
      </timeZone>
      <occurrenceType>ONCE</occurrenceType>
    </scheduling>
    <notification>
      <active>false</active>
    </notification>
    <target>
      <webApp><id>522066</id></webApp>
       <scannerTags>
           <set>
               <Tag>
                   <id>15415353311147</id>
               </Tag>
           </set>
       </scannerTags>
    </target>
    <profile><id>53483</id></profile>
  </WasScanSchedule>
</data>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
            <id>141147</id>
            <name>
                <![CDATA[Scheduled Scan With Pool of Internal
Scanners]]>
            </name>
            <owner>
                <id>1056860</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <active>false</active>
            <multi>false</multi>
            <type>VULNERABILITY</type>
            <target>
                <webApp>
                    <id>522065</id>
                    <name><![CDATA[My Web Application]]></name>
                    <url><![CDATA[http://mywebapp.com]]></url>
                </webApp>
                <scannerTags>
                    <set>
                        <Tag>
                            <id>8461819</id>
                        </Tag>
                    </set>
                </scannerTags>
            </target>
            <progressiveScanning>DEFAULT</progressiveScanning>
            <profile>
                <id>194283</id>
                <name>
                    <![CDATA[Initial WAS Options]]>
                </name>
            </profile>
            <scheduling>
                <startDate>2017-01-10T13:55:00Z</startDate>
                <timeZone>
                    <code>Europe/Istanbul</code>
                    <offset>+02:00</offset>
                </timeZone>
                <occurrenceType>ONCE</occurrenceType>
                <cancelAfterNHours>10</cancelAfterNHours>
            </scheduling>
            <notification>
                <active>false</active>
                <reschedule>false</reschedule>
            </notification>
            <launchedCount>0</launchedCount>
            <createdDate>2017-01-12T11:54:07Z</createdDate>
            <createdBy>
                <id>1056860</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
```

```
                <lastName><![CDATA[Doe]]></lastName>
            </createdBy>
            <updatedDate>2017-01-12T11:54:09Z</updatedDate>
            <updatedBy>
                <id>1056860</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </updatedBy>
            <sendMail>true</sendMail>
            <sendOneMail>false</sendOneMail>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```

Sample - Create or update schedule for progressive scanning

The user will be able to set progressiveScanning to ENABLED, DISABLED or DEFAULT, if progressiveScanning is enabled for the subscription. If this option is not set for a new schedule, the value DEFAULT is used.

API request

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

Request POST data

```
<ServiceRequest>
   <data>
      <WasScanSchedule>
            <name><![CDATA[Schedule with enabled
progressiveScanning]]></name>
            <type>VULNERABILITY</type>
            <active>false</active>
            <scheduling>
            <startDate>2019-01-30T12:40:27Z</startDate>
            <timeZone>
                  <code>Asia/Kolkata</code>
                  <offset>+05:30</offset>
            </timeZone>
            <occurrenceType>ONCE</occurrenceType>
            </scheduling>
            <notification>
                  <active>true</active>
                <delay>
                  <nb>1</nb>
                  <scale>DAY</scale>
                </delay>
                  <message><![CDATA[A scan is scheduled to start 
soon.]]></message>
            </notification>
            <target>
                  <webApps>
                        <set>
                           <WebApp><id>8389207</id></WebApp>
                        </set>
                  </webApps>
                  <scannerAppliance>
                        <type>EXTERNAL</type>
                  </scannerAppliance>
            </target>
            <progressiveScanning>ENABLED</progressiveScanning>
                  <profile>
                        <id>53483</id>
                  </profile>
            </WasScanSchedule>
      </data>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScanSchedule>
            <id>8831789</id>
            <name>
                <![CDATA[Schedule with enabled progressiveScanning]]>
            </name>
            <owner>
                <id>1056860</id>
                <username>user_john</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Doe]]>
                </lastName>
            </owner>
            <active>false</active>
            <multi>false</multi>
            <type>VULNERABILITY</type>
            <target>
                <webApp>
                    <id>8389207</id>
                    <name>
                        <![CDATA[My Web Application]]>
                    </name>
                    <url>
                     <![CDATA[http://mywebapp.com]]>
                    </url>
                </webApp>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
            </target>
            <progressiveScanning>ENABLED</progressiveScanning>
            <profile>
                <id>53483</id>
                <name>
                    <![CDATA[Scan OP]]>
                </name>
            </profile>
            <scheduling>
                <startDate>2019-01-30T12:40:00Z</startDate>
                <timeZone>
                    <code>Asia/Kolkata</code>
                    <offset>+05:30</offset>
                </timeZone>
                <occurrenceType>ONCE</occurrenceType>
            </scheduling>
            <notification>
                <active>true</active>
                <reschedule>false</reschedule>
                <delay>
                    <nb>1</nb>
                    <scale>DAY</scale>
                </delay>
                <message>
                    <![CDATA[A scan is scheduled to start soon.]]>
                </message>
            </notification>
            <launchedCount>0</launchedCount>
            <createdDate>2019-02-26T07:17:22Z</createdDate>
            <createdBy>
                <id>1056860</id>
                <username>user_john</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Doe]]>
                </lastName>
            </createdBy>
            <updatedDate>2019-02-26T07:17:22Z</updatedDate>
            <updatedBy>
                <id>1056860</id>
                <username>user_john</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Doe]]>
                </lastName>
            </updatedBy>
            <sendMail>true</sendMail>
            <sendOneMail>false</sendOneMail>
            <enableWAFAuth>false</enableWAFAuth>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```

If Progressive Scanning is not enabled for the subscription, the progressiveScanning element cannot be provided, otherwise an error will be returned.

XML response (error)

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

```
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.
0/was/wasscanschedule.xsd">
    <responseCode>INVALID_REQUEST</responseCode>
    <responseErrorDetails>
        <errorMessage>Progressive scanning is not enabled in your
subscription.</errorMessage>
        <errorResolution>Please check with your account manager to
enable this option.</errorResolution>
    </responseErrorDetails>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscanschedule.xsd

Create Schedules (Multiple)

/qps/rest/3.0/create/was/wasscanschedule

[POST]

You can schedule a Multi-Scan to run automatically, on a regular basis. This way you always have the most up-to-date security information in your account.

A Multi-Scan allows you to scan any number of web applications. This feature enables you to scan hundreds or even thousands of web applications you may have in your organization with granular insight into what scans are running and which ones are complete.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "Create WAS Schedule" permission. The output includes schedules in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. See Reference: WasScanSchedule for descriptions of these <WasScanSchedule> elements.

Click here for available operators

Parameter	Description
name	(text) Name of the schedule.
webApps.id or tags.id	(integer) The web applications to be scanned. webApps.id: Specify the web application ID to include it in the scan. tags.id: Specify the tag ID associated with the web applications to be scanned.

target.tags.excluded.option	(keyword: ALL or ANY) Decides which web applications should be excluded from the scan. ALL : Only the web applications associated with all the specified tags are excluded from the scan. ANY : Only the web applications associated with any of the specified tags are excluded from the scan.
target.tags.excluded.tagList.Tag.id	(integer) The web applications associated with the tag (identified by the specified tag ID) are excluded from the scan.
target.tags.included.option	(keyword: ALL or ANY) Decides which web applications should be excluded from the scan. ALL : Only the web applications associated with all the specified tags are excluded from the scan. ANY : Only the web applications associated with any of the specified tags are excluded from the scan.
target.tags.included.tagList.Tag.id	(integer) The web applications associated with the tag (identified by the specified tag ID) are included in the scan.
type	(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.
profile.id (integer)$_2$	(integer) The name of the option profile that includes scan settings. The service provides the profile “Initial WAS Options” and we recommend this to get started. `Example:`

```
<profile>

    <name>Initial WAS Options</name>

</profile>
```

startDate (date)	(date) The date when the schedule starts in UTC date/time format.
timeZone (text)	(text) The timezone in which the scan is scheduled in UTC date/time format.
occurrenceType	(keyword) The frequency of the scheduled scan : ONCE, DAILY, WEEKLY or MONTHLY.
notification	(boolean)A flag indicating whether email notification is enabled for scheduled scan.
reschedule	(boolean) Set this flag to reschedule the scan.
target.authRecordOption	(integer) Defines the authentication record to be used during the scan. Set to SPECIFIC -Always use the authRecord passed while launching the scan. Set to DEFAULT- Forces the use of the authRecord, if set, else fall back to the one passed in to the API while launching the scan.
target.profileOption	(keyword: ALL or ANY) Defines the option profile to be used during the scan.

	Set to SPECIFIC - Always use the optionProfile passed while launching the scan. Set to DEFAULT - Forces the use of the optionProfile if set, else fall back to the one passed in to the API while launching the scan.
target.scannerOption	(integer) Defines the scanner appliance to be used during the scan. Set to SPECIFIC - Always use the scanner passed while launching the scan Set to DEFAULT - Forces the use of the scanner if set, else fall back to the one passed in to the API while launching the scan.
target.randomizeScan	Allows the service to scan the selected web applications in random order. The randomness will help prevent network slowdowns and/or errors
target.scannerAppliance.type	(keyword: EXTERNAL or INTERNAL or scannerTags) Type of the scanner appliance to be used for the scan.
target.scannerAppliance.friendly Name	(text) Name of the scanner appliance being used for the scan.
cancelOption	set to DEFAULT - Forces the use of the target web app’s cancelScans option if set, else fall back to the one passed in to the API while launching the scan. set to SPECIFIC - Always use the cancel scan option passed while launching the scan.

sendMail	(boolean) Set to false to disable scan complete email notifications. `Example:<sendMail>false</sendMail>`
sendOneMail	(boolean) Set to true to send one email upon multi-scan completion. Set to false to send one email upon completion of each individual scan. `Example:<sendOneMail>true</sendOneMail>` Note: sendOneMail is valid only when sendMail = true for a multi-scan (multiple web applications being scanned). If sendMail is set to false, sendOneMail will be ignored.

sendMailFromAddressOption	Identifies the sender of the scan complete notifications. The valid values are: QUALYS_SUPPORT and OWNER. OWNER means the user whose account is used to create the schedule.
	`Example:<sendMailFromAddressOption>QUALYS_SUPPORT</sendMailFromAddressOption>`
	`Example:<sendMailFromAddressOption>OWNER</sendMailFromAddressOption>`
	To set this parameter, the sendMail parameter must be set to true. If the sendMail parameter is true, then sendMailFromAddressOption is by default set to QUALYS_SUPPORT. You can change the value of the parameter to OWNER.

enableWAFAuth	(boolean) Set to true to allow Qualys scanners to scan a web application through WAF. Note: Enabling this option enhances assessment and reporting of WAF-blocked vulnerabilities that are not yet fixed in WAS detections and reports.

1 The element target must have at least tags or web applications specified.

2 The element profile (text) is required unless the target has a default option profile.

Sample - Schedule a multi-scan

Let's schedule a multi-scan for two web applications by specifying the ID for the web applications.

API request

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <WasScanSchedule>
            <name>MultiSchedule_1497351121650</name>
            <type>VULNERABILITY</type>
            <active>false</active>
            <scheduling>
                <cancelAfterNHours>8</cancelAfterNHours>
                <startDate>2017-06-13T21:51:57Z</startDate>
                <timeZone>
                    <code>America/Vancouver</code>
                    <offset>-07:00</offset>
                </timeZone>
                <occurrenceType>WEEKLY</occurrenceType>
                <occurrence>
                    <weeklyOccurrence>
                        <everyNWeeks>2</everyNWeeks>
                        <occurrenceCount>20</occurrenceCount>
                        <onDays>
                            <WeekDay>SATURDAY</WeekDay>
                        </onDays>
                    </weeklyOccurrence>
                </occurrence>
            </scheduling>
            <notification>
                <active>true</active>
                <reschedule>true</reschedule>
                <delay>
                    <nb>1</nb>
                    <scale>DAY</scale>
                </delay>
                <message><![CDATA[A scan is scheduled to start 
soon.]]></message>
            </notification>
            <target>
                <webApps>
                    <set>
                        <WebApp>
                            <id>4331923</id>
                        </WebApp>
                        <WebApp>
                            <id>4331924</id>
                        </WebApp>
                    </set>
                </webApps>
                <webAppAuthRecord>
                    <id>583957</id>
                </webAppAuthRecord>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <cancelOption>SPECIFIC</cancelOption>
                <authRecordOption>DEFAULT</authRecordOption>
                <profileOption>SPECIFIC</profileOption>
                <scannerOption>DEFAULT</scannerOption>
                <randomizeScan>true</randomizeScan>
                <useDnsOverride>true</useDnsOverride>
            </target>
            <profile>
```

```
                <id>1071133</id>
            </profile>
        </WasScanSchedule>
    </data>
</ServiceRequest>
```

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/3.
0/
was/wasscanschedule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScanSchedule>
            <id>697193</id>
            <name><![CDATA[MultiSchedule_1497351121650]]></name>
            <owner>
                <id>2911477</id>
                <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
        </owner>
            <active>false</active>
            <multi>true</multi>
            <type>VULNERABILITY</type>
            <target>
                <webApps>
                    <list>
                        <WebApp>
                            <id>4331923</id>
                            <name><![CDATA[web app
1497351058103]]></name>
<url><![CDATA[http://www.example.com/cassium/xss/]]></url>
                        </WebApp>
                        <WebApp>
                            <id>4331924</id>
                            <name><![CDATA[web app
1497351100446]]></name>
<url><![CDATA[http://www.example.com/cassium/xss/]]></url>
                        </WebApp>
                    </list>
                </webApps>
                <webAppAuthRecord>
                    <id>583957</id>
                    <name><![CDATA[Form and
Server]149735111801]]></name>
                </webAppAuthRecord>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <cancelOption>SPECIFIC</cancelOption>
                <authRecordOption>DEFAULT</authRecordOption>
                <profileOption>SPECIFIC</profileOption>
                <scannerOption>DEFAULT</scannerOption>
                <randomizeScan>true</randomizeScan>
                <useDnsOverride>true</useDnsOverride>
            </target>
            <progressiveScanning>DEFAULT</progressiveScanning>
            <profile>
                <id>1071133</id>
                <name><![CDATA[My Option Profile - with defaults
1497351048931]]></name>
            </profile>
            <scheduling>
                <startDate>2017-06-13T21:51:00Z</startDate>
                <timeZone>
                    <code>America/Vancouver</code>
                    <offset>-07:00</offset>
                </timeZone>
                <occurrenceType>WEEKLY</occurrenceType>
                <occurrence>
                    <weeklyOccurrence>
                        <everyNWeeks>2</everyNWeeks>
                        <onDays>
                            <WeekDay>SATURDAY</WeekDay>
                        </onDays>
                        <occurrenceCount>20</occurrenceCount>
                    </weeklyOccurrence>
                </occurrence>
                <cancelAfterNHours>8</cancelAfterNHours>
            </scheduling>
            <notification>
                <active>true</active>
                <reschedule>true</reschedule>
                <delay>
                    <nb>1</nb>
                    <scale>DAY</scale>
                </delay>
                <message><![CDATA[A scan is scheduled to start
soon.]]></message>
            </notification>
            <launchedCount>0</launchedCount>
            <createdDate>2017-06-13T10:52:07Z</createdDate>
            <createdBy>
                <id>2911477</id>
                <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </createdBy>
            <updatedDate>2017-06-13T10:52:09Z</updatedDate>
            <updatedBy>
                <id>2911477</id>
                 <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </updatedBy>
            <sendMail>true</sendMail>
            <sendOneMail>false</sendOneMail>
            <enableWAFAuth>false</enableWAFAuth>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```

Sample - Schedule a multi-scan with some criteria

Let’s schedule a multi-scan for all the web applications that are associated with the tags specified in the request filter and configure scan completion notification to be sent after completion of the multi-scan.

API request

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

Request POST data

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

```
        <WasScanSchedule>
            <name>SampleSchedule</name>
            <type>VULNERABILITY</type>
            <active>false</active>
            <scheduling>
                <cancelAfterNHours>8</cancelAfterNHours>
                <startDate>2017-06-13T21:51:57Z</startDate>
                <timeZone>
                    <code>America/Vancouver</code>
                    <offset>-07:00</offset>
                </timeZone>
                <occurrenceType>WEEKLY</occurrenceType>
                <occurrence>
                    <weeklyOccurrence>
                        <everyNWeeks>2</everyNWeeks>
                        <occurrenceCount>20</occurrenceCount>
                        <onDays>
                            <WeekDay>SATURDAY</WeekDay>
                        </onDays>
                    </weeklyOccurrence>
                </occurrence>
            </scheduling>
            <notification>
                <active>true</active>
                <reschedule>true</reschedule>
                <delay>
                    <nb>1</nb>
                    <scale>DAY</scale>
                </delay>
                <message><![CDATA[A scan is scheduled to start 
soon.]]></message>
            </notification>
            <target>
            <tags>
                    <included>
                        <option>ALL</option>
                        <tagList>
                            <set>
                                <Tag>
                                    <id>12017424</id>
                                </Tag>
                                <Tag>
                                    <id>12017228</id>
                                </Tag>
                            </set>
                        </tagList>
                    </included>
                    <excluded>
                        <option>ANY</option>
                        <tagList>
                            <set>
                                <Tag>
                                    <id>12017228</id>
                                </Tag>
                            </set>
                        </tagList>
                    </excluded>
                </tags>
                <webAppAuthRecord>
                    <id>583957</id>
                </webAppAuthRecord>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <cancelOption>SPECIFIC</cancelOption>
                <authRecordOption>DEFAULT</authRecordOption>
                <profileOption>SPECIFIC</profileOption>
                <scannerOption>DEFAULT</scannerOption>
                <randomizeScan>true</randomizeScan>
                <useDnsOverride>true</useDnsOverride>
            </target>
            <profile>
                <id>1071133</id>
            </profile>
            <sendOneMail>false</sendOneMail>
        </WasScanSchedule>
    </data>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScanSchedule>
            <id>699795</id>
            <name>
                <![CDATA[Schedule a multi scan for multiple web 
apps]]>
            </name>
            <owner>
                <id>2911477</id>
                <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <active>false</active>
            <multi>true</multi>
            <type>VULNERABILITY</type>
            <target>
                <tags>
                    <included>
                        <option>ANY</option>
                        <tagList>
                            <list>
                                <Tag>
                                    <id>12017424</id>
                                </Tag>
                                <Tag>
                                    <id>12017228</id>
                                </Tag>
                            </list>
                        </tagList>
                    </included>
                    <excluded>
                        <option>ANY</option>
                        <tagList>
                            <list>
                                <Tag>
                                    <id>12017228</id>
                                </Tag>
                            </list>
                        </tagList>
                    </excluded>
                </tags>
                <webAppAuthRecord>
                    <id>583957</id>
                    <name>
                        <![CDATA[Form and Server]149735111801]]>
                    </name>
                </webAppAuthRecord>
                <scannerAppliance>
                    <type>EXTERNAL</type>
                </scannerAppliance>
                <cancelOption>SPECIFIC</cancelOption>
                <authRecordOption>DEFAULT</authRecordOption>
                <profileOption>SPECIFIC</profileOption>
                <scannerOption>DEFAULT</scannerOption>
                <randomizeScan>true</randomizeScan>
                <useDnsOverride>true</useDnsOverride>
            </target>
            <progressiveScanning>DEFAULT</progressiveScanning>
            <profile>
                <id>1071133</id>
                <name>
                    <![CDATA[My Option Profile - with defaults
1497351048931]]>
                </name>
            </profile>
            <scheduling>
                <startDate>2017-06-13T21:51:00Z</startDate>
                <timeZone>
                    <code>America/Vancouver</code>
                    <offset>-07:00</offset>
                </timeZone>
                <occurrenceType>WEEKLY</occurrenceType>
                <occurrence>
                    <weeklyOccurrence>
                        <everyNWeeks>2</everyNWeeks>
                        <onDays>
                            <WeekDay>SATURDAY</WeekDay>
                        </onDays>
                        <occurrenceCount>20</occurrenceCount>
                    </weeklyOccurrence>
                </occurrence>
                <cancelAfterNHours>8</cancelAfterNHours>
            </scheduling>
            <notification>
                <active>true</active>
                <reschedule>true</reschedule>
                <delay>
                    <nb>1</nb>
                    <scale>DAY</scale>
                </delay>
                <message>
                    <![CDATA[A scan is scheduled to start soon.]]>
                </message>
            </notification>
            <launchedCount>0</launchedCount>
            <createdDate>2017-06-15T09:19:09Z</createdDate>
            <createdBy>
                <id>2911477</id>
                 <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </createdBy>
            <updatedDate>2017-06-15T09:19:09Z</updatedDate>
            <updatedBy>
                <id>2911477</id>
                 <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </updatedBy>
            <sendMail>true</sendMail>
            <sendOneMail>false</sendOneMail>
            <enableWAFAuth>false</enableWAFAuth>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscanschedule.xsd

Update Schedule

/qps/rest/3.0/update/was/wasscanschedule/<id>

[POST]

Update a scheduled scan on a web application which is in the user’s scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and “Edit WAS Schedule”. Scan target must be within the user’s scope.

Input Parameters

The “id” (integer) element and the data to be updated in the schedule are required where “id” identifies a schedule. See Reference: WasScanSchedule for descriptions of all of the <WasScanSchedule> elements.

Click here for available operators

Sample - Update a schedule by enabling notification for the same

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscanschedule/
1688” < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <data>
    <WasScanSchedule>
      <notification>
        <active>true</active>
        <delay>
          <nb>4</nb>
          <scale>DAY</scale>
        </delay>
        <recipients>
```

```
          <set>
            <EmailAddress><![CDATA[name1@company.com]]></EmailAddress>

            <EmailAddress><![CDATA[name2@company.com]]></EmailAddress>

            <EmailAddress><![CDATA[name3@company.com]]></EmailAddress>

          </set>
        </recipients>
        <message><![CDATA[The schedule notification
message]]></message>
      </notification>
    </WasScanSchedule>
  </data>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>1688</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Sample - Update notification to reschedule

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscanschedule/
171425669" < file.xml
Note: "file.xml" contains the request POST data.
```

Request POST data

```
<ServiceRequest>
 <data>
```

```
   <WasScanSchedule>
<name><![CDATA[Update Notification to enable Reschedule]]></name>
 <notification>
        <active>true</active>
        <reschedule>true</reschedule>
        <delay>
          <nb>1</nb>
          <scale>DAY</scale>
        </delay>
        <message><![CDATA[A Qualys scan is scheduled to start
soon.]]></message>
      </notification>
   </WasScanSchedule>
 </data>
</ServiceRequest>
```

XML 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/3.0
/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>171425669</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Sample - Update schedule to configure scan completion notification

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/wasscanschedule/
171425669” < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
      <data>
```

```
         <WasScanSchedule>
            <name>Schedule with sendOneMail enabled</name>
            <sendMail>true</sendMail>
            <sendOneMail>true</sendOneMail>
         </WasScanSchedule>
      </data>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WasScanSchedule>
            <id>171425669</id>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscanschedule.xsd

Activate an Existing Schedule

/qps/rest/3.0/update/was/wasscanschedule/<id>

/qps/rest/3.0/activate/was/wasscanschedule/<id>

/qps/rest/3.0/activate/was/wasscanschedule/<filters>

[POST]

Activate one or more scheduled scans on web applications which are in the user’s scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and “Edit WAS Schedule”. Scan target must be within the user’s scope.

Input Parameters

The “id” (integer) element and the data to be updated in the schedule are required where “id” identifies a schedule. When multiple elements are specified, parameters are combined using a logical AND. See Reference: WasScanSchedule for descriptions of all of the <WasScanSchedule> elements.

Click here for available operators

Parameter	Description
id	(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.
name	(text) The user-defined schedule name (maximum 256 characters).
createdDate	(date) The date when the schedule was created in WAS, in UTC date/time format.

updatedDate	(date) The date when the schedule was created in WAS, in UTC date/time format.
type	(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.
webApp.name	(text) The name of the web application being scanned.
webApp.id	(integer) The ID of the web application being scanned.
owner.id	(text) ID associated with the owner who created the schedule.
active	(boolean) Indicates whether the schedule is active or not. True indicates active schedule.
invalid	(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.

Sample - Activate a schedule

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/activate/was/wasscanschedul
e/1688” < file.xml
```

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/3.
0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
```

```
      <id>1688</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Sample - Activate Multi Schedule using filters

API request

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

Request POST data

```
<ServiceRequest>
      <filters>
          <Criteria field="name"
operator="CONTAINS">Schedule</Criteria>
      </filters>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <WasScanSchedule>
            <id>701147</id>
        </WasScanSchedule>
        <WasScanSchedule>
            <id>701946</id>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscanschedule.xsd

Deactivate Schedule

/qps/rest/3.0/update/was/wasscanschedule/<id>

/qps/rest/3.0/deactivate/was/wasscanschedule/<id>

/qps/rest/3.0/deactivate/was/wasscanschedule/<filters>

[POST]

Deactivate one or more scheduled scans on web applications which are in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access" and "Edit WAS Schedule". Scan target must be within the user's scope.

Input Parameters

The "id" (integer) element and the data to be updated in the schedule are required where "id" identifies a schedule. When multiple elements are specified, parameters are combined using a logical AND. See Reference: WasScanSchedule for descriptions of all of the <WasScanSchedule> elements.

Click here for available operators

Parameter	Description
id	(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.
name	(text) The user-defined schedule name (maximum 256 characters).
createdDate	(date) The date when the schedule was created in WAS, in UTC date/time format.

updatedDate	(date) The date when the schedule was created in WAS, in UTC date/time format.
type	(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.
webApp.id	(integer) The ID of the web application being scanned.
webApp.name	(text) The name of the web application being scanned.
owner.id	(integer) ID associated with the owner who created the schedule.
active	(boolean) Indicates whether the schedule is active or not. True indicates active schedule.
invalid	(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.

Sample - Deactivate a schedule

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/deactivate/was/wasscansched
ule/1688" < file.xml
```

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/3.
0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
```

```
      <id>1688</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Sample - Deactivate Multi Schedule using filters

API request

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

Request POST data

```
<ServiceRequest>
      <filters>
          <Criteria field="name"
operator="CONTAINS">Schedule</Criteria>
      </filters>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <WasScanSchedule>
            <id>701147</id>
        </WasScanSchedule>
        <WasScanSchedule>
            <id>701946</id>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscanschedule.xsd

Delete Schedule

/qps/rest/3.0/delete/was/wasscanschedule/<id>

/qps/rest/3.0/delete/was/wasscanschedule/<filters>

[POST]

Delete scheduled scans on web applications which are in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access" and Delete WAS Schedule". Scan target must be within the user's scope.

Input Parameters

The "id" (integer) element and the data to be updated in the schedule are required where "id" identifies a schedule. When multiple elements are specified, parameters are combined using a logical AND. See Reference: WasScanSchedule for descriptions of all of the <WasScanSchedule> elements.

Click here for available operators

Parameter	Description
id	(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.
name	(text) The user-defined schedule name (maximum 256 characters).
createdDate	(date) The date when the schedule was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the schedule was created in WAS, in UTC date/time format.

type	(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.
webApp.name	(text) The name of the web application being scanned.
webApp.id	(integer) The ID of the web application being scanned.
owner.id	(integer) ID associated with the owner who created the schedule.
active	(boolean) Indicates whether the schedule is active or not. True indicates active schedule.
invalid	(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.

Sample - Delete single schedule

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/wasscanschedule/
1846”
Note: “file.xml” contains the request POST data.
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.cm/qps/xsd/3.0
/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>1846</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

Sample - Delete schedules matching criteria

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
       <Criteria field="active" operator="EQUALS">false</Criteria>
       <Criteria field="name" operator="CONTAINS">WEEKLY -</Criteria>
    </filters>
</ServiceRequest>
```

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/3.
0/was/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <data>
    <WasScanSchedule>
      <id>1747</id>
    </WasScanSchedule>
    <WasScanSchedule>
      <id>1768</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscanschedule.xsd

Download Schedule

/qps/rest/3.0/download/was/wasscanschedule/<id>

/qps/rest/3.0/download/was/wasscanschedule/<filters>

[POST]

Download scheduled scans on a web applications, which are in the user's scope, to iCalendar format and then import them into your favorite calendar application so you can access your schedules on the go. You can import your schedules into several calendars including Microsoft Outlook, Google Calendar and Apple iCal.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The schedule must be within the user's scope.

Input Parameters

The "id" (integer) element and the data to be updated in the schedule are required where "id" identifies a schedule. When multiple elements are specified, parameters are combined using a logical AND. See Reference: WasScanSchedule for descriptions of all of the <WasScanSchedule> elements.

Click here for available operators

Parameter	Description
id	(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.
name	(text) The user-defined schedule name (maximum 256 characters).
createdDate	(date) The date when the schedule was created in WAS, in UTC date/time format.

updatedDate	(date) The date when the schedule was created in WAS, in UTC date/time format.
type	(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.
webApp.name	(text) The name of the web application being scanned.
webApp.id	(integer) The ID of the web application being scanned.
owner.id	(integer) ID associated with the owner who created the schedule.
active	(boolean) Indicates whether the schedule is active or not. True indicates active schedule.
invalid	(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.

Sample - Download a single schedule

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/download/was/wasscanschedul
e/1846"
Note: “file.xml” contains the request POST data.
```

XML response

```
BEGIN:VCALENDAR
PRODID:-//Qualys Inc//WAS Product//EN
VERSION:2.0
CALSCALE:GREGORIAN
METHOD:PUBLISH
BEGIN:VTIMEZONE
TZID:America/Boise
```

```
TZURL:http://tzurl.org/zoneinfo/America/Boise
X-LIC-LOCATION:America/Boise
BEGIN:DAYLIGHT
TZOFFSETFROM:-0700
TZOFFSETTO:-0600
TZNAME:MDT
DTSTART:20070311T020000
RRULE:FREQ=YEARLY;BYMONTH=3;BYDAY=2SU
END:DAYLIGHT
BEGIN:STANDARD
TZOFFSETFROM:-0600
TZOFFSETTO:-0700
TZNAME:MST
DTSTART:20071104T020000
RRULE:FREQ=YEARLY;BYMONTH=11;BYDAY=1SU
END:STANDARD
BEGIN:STANDARD
TZOFFSETFROM:-074449
TZOFFSETTO:-0800
TZNAME:PST
DTSTART:18831118T121511
END:STANDARD
BEGIN:DAYLIGHT
...
CREATED:20181128T204534Z
LAST-MODIFIED:20181128T210007Z
SEQUENCE:0
STATUS:CONFIRMED
TRANSP:TRANSPARENT
END:VEVENT
END:VCALENDAR
```

XSD

<platform API server>/qps/xsd/3.0/was/wasscanschedule.xsd

Reference: Schedule

The <WasScanSchedule> element includes sub elements used to define a schedule. A reference of these elements is provided below. An asterisk * indicates a complex element.

Parameter	Description
id	(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.
name	(text) The user-defined schedule name (maximum 256 characters).
owner.id	(integer) ID associated with the owner who created the schedule.
createdDate	(date) The date when the schedule was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the schedule was created in WAS, in UTC date/time format.
type	(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.
webApp.name	(text) The name of the web application being scanned.
webApp.id	(integer) The ID of the web application being scanned.
webApp.tags (with operator="NONE")	Tags associated with the web application being scanned.
webApp.tags.id	(integer) ID of the tag applied to the web application being scanned.

invalid	(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.
lastScan (with operation="NONE")	(boolean) Indicates if the last scan was performed or not. True indicates that the last scan was performed.
lastScan.launchedDate	(date) Date when the last scan was launched on the web application, in UTC date/time format.
lastScan.status	(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR, CANCELED)
multi (Boolean)	(boolean) Indicates if the scheduled scan is single scan or multiple scan.

Reference: WasScanSchedule

The <WasScanSchedule> element includes sub elements used to define a web application scan schedule. A reference of these elements is provided below. An asterisk * indicates a complex element.

Parameter	Description
id	(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request (details, activate, deactivate).

owner	(text)The user who owns the schedule. User properties include user ID, user login, first and last name.
	`Example:`
	`<owner>`
	`  <id>123056</id>`
	`  <username>username</username>`
	`  <firstName><![CDATA[John]]></firstName>`
	`  <lastName><![CDATA[Smith]]></lastName>`
	`</owner>`

active (Boolean)	The schedule is active: true or false.
launchedCount (integer)	The number of times the scan has been launched.
nextLaunchDate (date)	The next launch date and time in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).
target* (for single web application)	(text) The target of the scan. <webApp> is the target web application. `<scannerAppliance>` - type (keyword) is set to INTERNAL for a scanner appliance, or EXTERNAL for external scanners or

scannerTags for assigning multiple scanner appliances grouped by asset tag. If the type is INTERNAL, friendlyName (text) is the user-defined appliance name.

<cancelOption> set to DEFAULT - Forces the use of the target web app's cancelScans option if set, else fall back to the one passed in to the API with the schedule settings.

<cancelOption> set to SPECIFIC - Always use the cancel scan option passed with the schedule settings.

Example: target.webApp is required

```
<target>
  <webApp>
    <id>324265</id>
    <name><![CDATA[Merchant Site]]></name>
    <url><![CDATA[http://url]]></url>
  </webApp>
  <scannerAppliance>
    <type>INTERNAL</type>
    <friendlyName><![CDATA[name]]></friendl
yName>
  </scannerAppliance>
  <cancelOption>SPECIFIC</cancelOption>
</target>
```

target* (for multiple web application)	<cancelOption> set to DEFAULT - Forces the use of the target web app's cancelScans option if set, else fall back to the one passed in to the API while launching the scan. <cancelOption> set to SPECIFIC - Always use the cancel scan option passed while launching the scan. <target.authRecordOption> set to SPECIFIC -Always use the authRecord passed while launching the scan

`<target.authRecordOption>` set to DEFAULT-Forces the use of the authRecord, if set, else fall back to the one passed in to the API while launching the scan.

`<target.profileOption>` set to SPECIFIC-Always use the optionProfile passed while launching the scan

`<target.profileOption>` set to DEFAULT-Forces the use of the optionProfile if set, else fall back to the one passed in to the API while launching the scan.

`<target.scannerOption>` set to SPECIFIC-Always use the scanner passed while launching the scan

`<target.scannerOption>` set to DEFAULTForces the use of the scanner if set, else fall back to the one passed in to the API while launching the scan.

`<target.randomizeScan>` (Boolean) - Set to true to scan the selected web applications in random order. Set to false to scan the selected web application in sequential order.

`target.tags` (For MultiScan)--

---target.tags.included.option(ALL/ANY) is required,

---target.tags.included.tagList is required, only <set> is allowed for target.tags.included.tagList.

--- target.tags.included.tagList.set.Tag.id is required and should be valid

---Only target.tags.exclusive is not allowed, it must be with target.tags.inclusive

---If target.tags.excluded is present, all the above rules are applicable to it

```
Example: Either target.webApps or
target.tags is required and these are
mutually exclusive.
target.webApps (For MultiScan)-
Only <set> is allowed for target.webApps
 <webApps>
   <set>
      <WebApp>
         <id>4330527</id>
      </WebApp>
      <WebApp>
         <id>4330327</id>
      </WebApp>
   </set>
  </webApps>
target.tags (For MultiScan)-
<tags>
     <included>
       <option>ALL</option>
         <tagList>
           <set>
             <Tag><id>12017424</id></Tag>
             <Tag><id>12017228</id></Tag>
           </set>
         </tagList>
         </included>
         <excluded>
            <option>ANY</option>
                <tagList>
                   <set>
                     <Tag><id>12017228</id>
</Tag>
                 </set>
                </tagList>
         </excluded>
   </tags>
```

profile.id

(integer) The name of the option profile that includes scan settings. The service provides the

	profile “Initial WAS Options” and we recommend this to get started.
	
	`Example:`
	`<profile>`
	`    <name>Initial WAS Options</name>`
	`</profile>`

proxy.id	(integer) The proxy for scanning the target web application.
	
	`Example:`
	`<proxy>`
	`   <id>12345</id>`
	`</proxy>`

dnsOverride.id	(integer) The DNS override record for scanning the target web application.
	
	`Example:`
	`<dnsOverride>`
	`   <id>67890</id>`
	`</dnsOverride>`

createdDate (date)	The schedule creation date and time in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

createdBy*	The user who created the schedule.
	
	`Example:`
	`<createdBy>`
	`  <id>123056</id>`
	`  <username>username</username>`
	`  <firstName><![CDATA[John]]></firstName>`
	`  <lastName><![CDATA[Smith]]></lastName>`
	`</createdBy>`

updatedDate (date)	The date and time of the most recent update of the schedule in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

updatedBy*

The user who updated the schedule.

```
Example:
<updatedBy>
  <id>123056</id>
  <username>username</username>
  <firstName><![CDATA[John]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</updatedBy>
```

scheduling*

The schedule settings.

<doNotCancel> is to run scan until it completes, or the maximum scan time is reached. This option can be set to true. If you want to cancel scan automatically after some period of time - after a number of hours, or at a specific time, use one of the following options:

<cancelAfterNHours> is the number of hours after which the scan task will be cancelled.

<cancelTime> is the time at which a scan will be cancelled.

Note: The three tags — <doNotCancel>, <cancelAfterNHours>, and <cancelTime> are mutually exclusive.

<startDate> is the date and time the scan will begin.

<timeZone> is the time zone that applies to the schedule.

<occurrenceType> defines frequency of the task: ONCE, DAILY, WEEKLY or MONTHLY.

Example of single occurrence scan with the <doNotCancel> option:

```
<scheduling>
    <doNotCancel>true</doNotCancel>
    <timeZone>
```

```
        <code>Europe/Paris</code>
    </timeZone>
    <occurrenceType>ONCE</occurrenceType>
</scheduling>
```

Example of weekly scan with the <cancelAfterNHours> option:

```
<scheduling>
  <cancelAfterNHours>11</cancelAfterNHours>
  <startDate>2017-02-
02T10:10:00Z</startDate>
   <timeZone>
     <code>Europe/Paris</code>
   </timeZone>
   <occurrenceType>WEEKLY</occurrenceType>
  <occurrence>
    <weeklyOccurrence>
      <everyNWeeks>2</everyNWeeks>
       <occurrenceCount>20</occurrenceCount
>
       <onDays>
        <WeekDay>MONDAY</WeekDay>
        <WeekDay>SATURDAY</WeekDay>
         <WeekDay>SUNDAY</WeekDay>
      </onDays>
    </weeklyOccurrence>
  </occurrence>
</scheduling>
```

Example of single occurrence scan with the <cancelTime> option:

```
<scheduling>
    <startDate>2017-02-
02T10:10:00Z</startDate>
    <cancelTime>11:15</cancelTime>
    <timeZone>
        <code>Europe/Paris</code>
    </timeZone>
    <occurrenceType>ONCE</occurrenceType>
</scheduling>
```

notification*

The notification settings.

- <active> indicates whether notification is enabled.

- <delay> indicates when the notification will be sent as number of days, hours, or minutes before the scan.

- <scale> indicates the delay unit: DAY, HOUR or MINUTE.

- <fromAddressOption> identifies the sender of the notification. The valid values for the tag are: QUALYS_SUPPORT and OWNER. OWNER means the user whose account is used to create the schedule. If you do not specify this tag, then by default the QUALYS_SUPPORT value is sent in the request for this tag.

```
<fromAddressOption>QUALYS_SUPPORT
</fromAddressOption>
<fromAddressOption>OWNER</fromAddressOption
>
```

- <recipients> identifies the email addresses of the notification recipients. <message> is the text of the notification message.

Example:

```
<notification>
  <active>true</active>
  <delay>
    <nb>1</nb>
    <scale>DAY</scale>
  </delay>
  <fromAddressOption>OWNER</fromAddressOpti
on>
  <recipients>
    <set>
<EmailAddress><![CDATA[1@a.com]]></EmailAdd
ress>
```

	`<EmailAddress><![CDATA[2@a.com]]></EmailAddress>`
	`    </set>`
	`  </recipients>`
	`  <message><![CDATA[The message]]></message>`
	`</notification>`

sendMail	(boolean) Set to false to disable scan complete email notifications. `Example:<sendMail>false</sendMail>`
sendOneMail	(boolean) Set to true to send one email upon multi-scan completion. Set to false to send one email upon completion of each individual scan. `Example:<sendOneMail>true</sendOneMail>` Note: sendOneMail is valid only when sendMail = true for a multi-scan (multiple web applications being scanned). If sendMail is set to false, sendOneMail will be ignored.

sendMailFromAddressOption	Identifies the sender of the scan complete notifications. The valid values are: QUALYS_SUPPORT and OWNER. OWNER means the user whose account is used to create the schedule.
	
	`Example:<sendMailFromAddressOption>QUALYS_SUPPORT</sendMailFromAddressOption>`
	`Example:<sendMailFromAddressOption>OWNER </sendMailFromAddressOption>`
	
	To set this parameter, the sendMail parameter must be set to true. If the sendMail parameter is true, then sendMailFromAddressOption is by default set to QUALYS_SUPPORT. You can change the value of the parameter to OWNER.

Reports

Report Count

/qps/rest/3.0/count/was/report

[GET] [POST]

Returns the total number of reports in the user's scope.

Permissions required User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes reports in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. See Reference: Report for descriptions of these <Report> elements

Click here for available operators

Parameter	Description
id	(integer) The report ID. This element is assigned by the service and is required for a certain type of request (details, status, update, delete, send or download).
name	(text) A report name (maximum 256 characters). Applies to all reports.
tags.id	(integer) ID of the tag associated with the report.
tags.name	(text) Name of the tag associated with the report.

creationDate	(date) The date when the report was created in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).
type	(keyword) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT, DATALIST_REPORT.
format	(keyword) The format of the report, one of: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, POWERPOINT, CSV, CSV_V2, XML, WORD.
status	(keyword) The status of the report: RUNNING, ERROR or COMPLETE.

Sample - Get count of reports in user's account

Return the number (count) of all reports in the user's scope.

API request

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

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/3.
0/was/report.xsd"
<ServiceResponse>
  <count>12</count>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

Sample - Get count of reports with a criteria

Return the number (count) reports with an ID that includes 1302 and 1303.

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="IN">1302, 1303</Criteria>
    </filters>
</ServiceRequest>
```

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/3.
0/was/report.xsd"
<ServiceResponse>
  <count>1</count>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Search Report

/qps/rest/3.0/search/was/report

[POST]

Returns a list of reports which are in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes reports in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. See Reference: Report for descriptions of these <Report> elements

Click here for available operators

Parameter	Description
id	(integer) The report ID. This element is assigned by the service and is required for a certain type of request (details, status, update, delete, send or download).
name	(text) A report name (maximum 256 characters). Applies to all reports.
tags.id	(integer) ID of the tag associated with the report.
tags.name	(text) Name of the tag associated with the report.
creationDate	(date) The date when the report was created in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).
type	(keyword) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT,

	WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT, DATALIST_REPORT.
format	(keyword) The format of the report, one of: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, POWERPOINT, CSV, CSV_V2, XML, WORD.
status	(keyword) The status of the report: RUNNING, ERROR or COMPLETE.

Sample - Search reports (no criteria)

Let us view a list of all reports in the user’s scope.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/search/was/report"
```

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/3.
0/was/report.xsd"
<ServiceResponse>
  <count>3</count>
  <data>
    <list>
      <Report>
        <id>1393</id>
        <name><![CDATA[Web Application Report 1]]></name>
        <type>WAS_WEBAPP_REPORT</type>
        <format>PDF</format>
        <status>COMPLETE</status>
        <size>2244667</size>
        <creationDate>2017-11-25T10:20:06Z</creationDate>
        <tags>
          <count>0</count>
        </tags>
        <owner>
          <id>123056</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </owner>
      </Report>
      <Report>
        <id>1394</id>
        <name><![CDATA[Web Application Report 2]]></name>
        <type>WAS_WEBAPP_REPORT</type>
        <format>PDF</format>
        <status>COMPLETE</status>
        <size>124578</size>
        <creationDate>2017-11-25T10:21:25Z</creationDate>
        <tags>
          <count>0</count>
        </tags>
        <owner>
          <id>123056</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </owner>
      </Report>
      <Report>
        <id>1282</id>
        <name><![CDATA[Web Application Report 3]]></name>
        <type>WAS_WEBAPP_REPORT</type>
        <format>PDF</format>
        <status>COMPLETE</status>
        <size>12341234</size>
        <creationDate>2017-11-24T00:00:00Z</creationDate>
        <tags>
          <count>0</count>
        </tags>
        <owner>
          <id>123056</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </owner>
      </Report>
    </list>
  </data>
  <isDone>true</isDone>
  <responseCode>SUCCESS</responseCode>
  <responseErrorDetails>
```

```
    <internalErrorCodeId>0</internalErrorCodeId>
  </responseErrorDetails>
</ServiceResponse>
```

Sample - Search for a particular report

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="tags.id" operator="EQUALS">99511</Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/was/report.xsd"
<ServiceResponse>
  <count>1</count>
  <data>
    <list>
      <Report>
        <id>1302</id>
        <name><![CDATA[Web Application Report 2]]></name>
        <type>WAS_WEBAPP_REPORT</type>
        <format>PDF_ENCRYPTED</format>
        <status>COMPLETE</status>
        <size>2244667</size>
        <creationDate>2017-11-24T00:00:00Z</creationDate>
        <tags>
          <count>1</count>
        </tags>
        <distributionList>
          <count>12</count>
        </distributionList>
        <owner>
          <id>123056</id>
          <username>username</username>
          <firstName><![CDATA[John]]></firstName>
          <lastName><![CDATA[Smith]]></lastName>
        </owner>
      </Report>
    </list>
  </data>
  <isDone>true</isDone>
  <responseCode>SUCCESS</responseCode>
  <responseErrorDetails>
    <internalErrorCodeId>0</internalErrorCodeId>
  </responseErrorDetails>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Get Report Details

/qps/rest/3.0/get/was/report/<id>

[GET]

View details for a report which is in the user's scope. Want to find a report ID to use as input? See Search reports.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes reports in the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies the report.

Click here for available operators

Sample - View details of a report

Let us view details for a report with ID 1302.

API request

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

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/3.
0/was/report.xsd"
<ServiceResponse>
  <count>1</count>
  <data>
    <Report>
      <id>1302</id>
      <name><![CDATA[Web Application Report 2]]></name>
      <type>WAS_WEBAPP_REPORT</type>
      <format>PDF_ENCRYPTED</format>
      <status>COMPLETE</status>
      <size>2244667</size>
      <creationDate>2018-11-24T00:00:00Z</creationDate>
      <lastDownloadDate>2018-11-09T00:00:00Z</lastDownloadDate>
      <downloadCount>1</downloadCount>
      <tags>
        <count>2</count>
        <list>
          <Tag>
            <id>99509</id>
            <name><![CDATA[Tag 1]]></name>
          </Tag>
          <Tag>
            <id>99510</id>
            <name><![CDATA[Tag 2]]></name>
          </Tag>
        </list>
      </tags>
      <distributionList>
        <count>2</count>
        <list>
<EmailAddress><![CDATA[email1@company.com]]></EmailAddress>
<EmailAddress><![CDATA[email2@company.com]]></EmailAddress>
        </list>
      </distributionList>
      <owner>
        <id>123056</id>
        <username>username</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
    </Report>
  </data>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Get Report Status

/qps/rest/3.0/status/was/report/<id>

[GET]

Retrieve the status of a report which is in the user's scope. Want to find a report ID to use as input? See Search reports.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes reports in the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies the report.

Click here for available operators

Sample - Get report status of a particular report

Let us view details for report with ID 1302.

API request

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

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/3.
0/was/report.xsd"
<ServiceResponse>
  <count>1</count>
  <data>
    <Report>
      <id>1302</id>
      <status>COMPLETE</status>
    </Report>
  </data>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Download Report

/qps/rest/3.0/download/was/report/<id>

[GET]

Download a report which is in the user's scope. Want to find a report ID to use as input? See Search reports.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes reports in the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies the report.

Click here for available operators

Sample - Download a report

Let us view download a report with ID 1302.

API request

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

XML response

Report ID 1302 will be downloaded in the format in which it was generated.

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Send Encrypted PDF Report

/qps/rest/3.0/send/was/report/<id>

[POST]

Send an encrypted PDF report, which is in the user’s scope, to a distribution list.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and “Distribute Report” permission. The output includes reports in the user's scope.

Input Parameters

The elements “id” (integer) and “distributionList” (text) are required, where “id” identifies a report and “distributionList” identifies the email addresses of the report recipients.

Click here for available operators

Sample - Send Encrypted PDF Report

Let us send an encrypted PDF report to a distribution list.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
@data-binary
"https://qualysapi.qualys.com/qps/rest/3.0/send/was/report/1302" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <data>
    <Report>
    <distributionList>
    <add>
      <EmailAddress><![CDATA[email1@abc.com]]></EmailAddress>
      <EmailAddress><![CDATA[email2@abc.com]]></EmailAddress>
    </add>
    </distributionList>
  </Report>
 </data>
</ServiceRequest>
```

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/3.
0/was/report.xsd"
<ServiceResponse>
  <count>1</count>
  <data>
    <Report>
      <id>1302</id>
    </Report>
  </data>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report .xsd

Update Report

/qps/rest/3.0/update/was/report/<id>

[POST]

Update the tags assigned to a report which is in the user’s scope. Want to find a report ID to use as input? See Search reports.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "Edit Report” permission. The output includes reports in the user's scope.

Input Parameters

The elements “id” (integer) and “tags” (complex element) are required, where “id” identifies a report and “tags” identifies tags to be added or removed.

The element “showPatched” can be set to filter the report to include/not include findings with virtual patches. Applies to Web Application Report and Scan Report. This filter can be set to:

SHOW_ONLY - show patched findings only

SHOW_BOTH - show patched & unpatched findings (default)

SHOW_NONE - show unpatched findings only

Click here for available operators

Sample - Update a report - add a tag

Let us update the a report with ID 1304 by tagging the report.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/report/1304" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <data>
        <Report>
            <tags>
                <set>
                    <Tag>
                        <id>99509</id>
                    </Tag>
                    <Tag>
                        <id>99510</id>
                    </Tag>
                </set>
            </tags>
        </Report>
    </data>
</ServiceRequest>
```

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/3.
0/was/report.xsd"
<ServiceResponse>
  <count>1</count>
  <data>
    <Report>
      <id>1304</id>
    </Report>
  </data>
  <responseCode>SUCCESS</responseCode>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Delete Report

/qps/rest/3.0/delete/was/report/<id>

/qps/rest/3.0/delete/was/report

[POST]

Delete a report which is in the user’s scope. Want to find a report ID to use as input? See Search reports.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "Delete Report” permission. The output includes reports in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. To delete one report by the report ID, the id element is required. the other elements listed below are used to delete reports based on filters. See Reference: Report for descriptions of these <Report> elements.

Click here for available operators

Parameter	Description
id	(integer) The report ID. This element is assigned by the service and is required for a certain type of request (details, status, update, delete, send or download).
name	(text) A report name (maximum 256 characters). Applies to all reports.
tags.id	(integer) ID of the tag associated with the report.
tags.name	(text) Name of the tag associated with the report.
creationDate	(date) The date when the report was created in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

type	(keyword) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT, DATALIST_REPORT.
format	(keyword) The format of the report, one of: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, POWERPOINT, CSV, XML, WORD.
status	(keyword) The status of the report: RUNNING, ERROR or COMPLETE.

Sample - Delete a single report

Let us delete report with the ID 6333.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/report/6333"
```

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/3.
0/was/report.xsd">
 <responseCode>SUCCESS</responseCode>
 <count>1</count>
 <data>
   <Report>
     <id>6333</id>
   </Report>
 </data>
```

Sample - Delete reports - criteria

Let us delete reports matching one or both of these criteria: 1) reports with names that contain the string “to be deleted”, and 2) reports that are completed (having the status COMPLETED).

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">to be
deleted</Criteria>
    <Criteria field="status" operator="EQUALS">COMPLETE</Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/was/report.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Report>
      <id>1542</id>
    </Report>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Report Creation

Create Report

/qps/rest/3.0/create/was/report

[POST]

Using the Report Creation API you can create different types of report: Web Application Report, Scan Report, Scorecard Report, Catalog Report.

Note: You can generate a report without the template ID or display and filter information. In such a case, we will use the default template based on the type of the report to generate the report.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and “Create Report”.

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Web Application Report

/qps/rest/3.0/create/was/report

[POST]

Using the Report Creation API you can create the Web Application Report.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and “Create Report”.

Note: Report creation may sometimes fail if the report is created for large number of web applications. To avoid such failures, we have now categorized report creation as per the number of web applications being included in the report. For web applications less than or equal to 500, you can create the report. But if the number of web applications exceeds 500, report cannot be created and error message is displayed in such cases.

The categorization is as follows:

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

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
name	(text) Name of the report.

	Note: Generating a report without template will allow you to assign a name to the report. If you use template during report generation, the name you provide in the request is ignored and the template name is assigned to the report.
type	(keyword) Type of the report, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT
format	(keyword) Report format, one of: WORD, HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, CSV, CSV_V2, XML, POWERPOINT
template.id	(integer) The template ID. This element is assigned by the system and is required for a certain type of request.
config*(1)	The "config" element must have one and only one of these child elements: webAppReport, scanReport, catalogReport or scorecardReport. Refer to Reference: Report for more details.
tags.id	(integer) ID of the tag associated with the web application.
password	(text) The password for a PDF encrypted report.

Sample - Create web app report - minimum criteria

Let us create a web application report in encrypted PDF format, setting both tags and web applications for the target.

API request

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

Request POST data

```
<ServiceRequest>
  <data>
     <Report>
            <name><![CDATA[API Web Application Report]]></name>
            <description><![CDATA[PDF WebApp report]]></description>
            <format>PDF</format>
            <type>WAS_WEBAPP_REPORT</type>
            <config>
            <webAppReport>
           <target>
          <webapps>
       <WebApp><id>8223303</id></WebApp>
 </webapps>
 </target>
 </webAppReport>
 </config>
 </Report>
  </data>
</ServiceRequest>
```

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/3.
0/was/report.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
    <data>
        <Report>
            <id>1085046</id>
        </Report>
    </data>
</ServiceResponse>
```

Sample - Create a web application report - use tags as target

Let us create a web application report using tags to add web applications as target for the report.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
```

```
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/report" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <data>
        <Report>
            <name><![CDATA[Web App Report]]></name>
            <format>PDF</format>
            <type>WAS_WEBAPP_REPORT</type>
            <config>
                <webAppReport>
                    <target>
                        <tags>
                            <included>
                                <option>ALL</option>
                                <tagList>
                                    <Tag>
                                        <id>12008216</id>
                                    </Tag>
                                 </tagList>
                            </included>
                            <excluded>
                                <option>ANY</option>
                                <tagList>
                                    <Tag>
                                        <id>12008219</id>
                                    </Tag>
</tagList>
                            </excluded>
                        </tags>
                    </target>
                </webAppReport>
            </config>
        </Report>
    </data>
</ServiceRequest>
```

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/3.
0/
was/report.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Report><id>981654</id>
        </Report>
    </data>
</ServiceResponse>
```

Sample - Create a web application report using report template

Let's generate a web application report in PDF format using a specific template (identified by its template ID).

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <Report>
          <name>Web_App_Report</name>
           <description><![CDATA[A web application
report]]></description>
            <type>WAS_WEBAPP_REPORT</type>
            <format>PDF</format>
            <config>
                <webAppReport>
                   <target>
                        <tags>
                            <included>
                                <option>ALL</option>
                                <tagList>
                                    <Tag>
                                    <id>12001856</id>
                                    </Tag>
                                </tagList>
                            </included>
                            <excluded>
                                <option>ANY</option>
                                <tagList>
                                    <Tag>
                                    <id>12001856</id>
                                    </Tag>
                                </tagList>
                            </excluded>
                        </tags>
                    </target>
                </webAppReport>
            </config>
            <template>
                <id>876048</id>
            </template>
        </Report>
    </data>
</ServiceRequest>
```

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/3.
0/
was/report.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Report>
            <id>973056</id>
        </Report>
    </data>
</ServiceResponse>
```

Sample - Create a web application report using CSV_V2 format

Let’s generate a web application report in CSV_V2 format.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
```

```
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/report/" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
   <data>
      <Report>
         <name><![CDATA[Web Application Report for Servers]]></name>
         <format>CSV_V2</format>
         <template>
            <id>46440</id>
         </template>
         <config>
            <webAppReport>
               <target>
                  <webapps>
                     <WebApp>
                        <id>470281</id>
                     </WebApp>
                  </webapps>
               </target>
            </webAppReport>
         </config>
      </Report>
   </data>
</ServiceRequest>
```

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/3.
0/was/report.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Report>
            <id>214158</id>
        </Report>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Scan Report

/qps/rest/3.0/create/was/report

[POST]

Using the Report Creation API you can create the Scan Report. A scan report shows you the results of scans on a particular web application.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and “Create Report”.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. The element “target” is required and at least one “scans” child element is required. For details, refer to Reference: Report Creation.

Click here for available operators

Parameter	Description
target.scans	(WasScan) The web applications to be scanned.
filters.searchlists	(SearchList) Number of search lists to report on vulnerabilities in those lists. If no search lists are selected, the report will include all findings.
filters.url	(text) Number of URLs of the web applications to being scanned.
filters.status	(ScanFindingStatus) Select status of vulnerabilities to be included in this report: New, Active, Re-opened, Fixed, Protected.

filters.remediation.showPatched	(keyword) Specify the filter to include ignored or patched findings (vulnerabilities and sensitive content) in this report. Show patched filter: SHOW_ONLY, SHOW_NONE, SHOW_BOTH - default.
filters.remediation.ignoredReasons	(keyword) The reason to ignore a finding: FALSE_POSITIVE, RISK_ACCEPTED, NOT_APPLICABLE.
display.contents	(ScanAppReportContent) The report content: Description, Summary, Results, Individual Records, Details, AllResults, Appendix, Severity Levels.
display.graphs	(ScanAppReportGraph) The graphs to be included in the report: Vulnerabilities by severity, Vulnerabilities by status, Vulnerabilities by group, Sensitive contents by group, Vulnerabilities by OWASP, Vulnerabilities by WASC, Most vulnerable URLs.
display.groups	(ScanAppReportGroup) The group category to be included in the report: URL, OWASP, WASC, State, Category, QID, Group.
display.options	(rawLevels) (Urgent), 4 (Critical), 3 (Serious), 2 (Medium), 1 (Minimal)
filters.remediation.showIgnored	(boolean) Specify if you wish to include ignored or patched findings.
format	(keyword) Report format, one of: WORD, HTML_ZIPPED, HTML_BASE64, PDF,

PDF_ENCRYPTED, CSV, CSV_V2,
XML, POWERPOINT

Sample - Create a scan report

Let us create a scan report in HTML ZIPPED format, selecting a single scan for the target.

API request

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

Request POST data

```
<ServiceRequest>
  <data>
    <Report>
      <name><![CDATA[with all parameters HTML_ZIPPED]]></name>
      <description><![CDATA[A simple scan report]]></description>
      <format>HTML_ZIPPED</format>
      <type>WAS_SCAN_REPORT</type>
      <config>
        <scanReport>
          <target>
            <scans>
              <WasScan>
                <id>104268</id>
              </WasScan>
            </scans>
          </target>
          <display>
            <contents>
              <ScanReportContent>DESCRIPTION</ScanReportContent>
              <ScanReportContent>SUMMARY</ScanReportContent>
              <ScanReportContent>GRAPHS</ScanReportContent>
              <ScanReportContent>RESULTS</ScanReportContent>
          <ScanReportContent>INDIVIDUAL_RECORDS</ScanReportContent>
              <ScanReportContent>RECORD_DETAILS</ScanReportContent>
              <ScanReportContent>ALL_RESULTS</ScanReportContent>
              <ScanReportContent>APPENDIX</ScanReportContent>
            </contents>
            <graphs>
              <ScanReportGraph>VULNERABILITIES_BY_SEVERITY</ScanReport
Graph>
              <ScanReportGraph>VULNERABILITIES_BY_GROUP</ScanReportGra
ph>
              <ScanReportGraph>VULNERABILITIES_BY_OWASP</ScanReportGra
ph>
              <ScanReportGraph>VULNERABILITIES_BY_WASC</ScanReportGrap
h>
              <ScanReportGraph>SENSITIVE_CONTENTS_BY_GROUP</ScanReport
Graph>
            </graphs>
            <groups>
              <ScanReportGroup>URL</ScanReportGroup>
              <ScanReportGroup>GROUP</ScanReportGroup>
              <ScanReportGroup>OWASP</ScanReportGroup>
              <ScanReportGroup>WASC</ScanReportGroup>
              <ScanReportGroup>STATUS</ScanReportGroup>
              <ScanReportGroup>CATEGORY</ScanReportGroup>
              <ScanReportGroup>QID</ScanReportGroup>
            </groups>
            <options>
              <rawLevels>true</rawLevels>
            </options>
          </display>
          <filters>
            <searchlists>
              <SearchList>
                <id>43147</id>
              </SearchList>
            </searchlists>
            <url>http://www.mysite.com/help.html</url>
            <status>
              <ScanFindingStatus>NEW</ScanFindingStatus>
              <ScanFindingStatus>ACTIVE</ScanFindingStatus>
              <ScanFindingStatus>REOPENED</ScanFindingStatus>
              <ScanFindingStatus>FIXED</ScanFindingStatus>
            </status>
          </filters>
        </scanReport>
      </config>
    </Report>
  </data>
```

```
</ServiceRequest>
```

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/3.
0/was/report.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Report>
      <id>3629</id>
    </Report>
  </data>
</ServiceResponse>
```

Sample - Create a scan report with remediation filter options

Let us create a scan report with remediation filter options to either include ignored findings.

API request

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

Request POST data

```
<ServiceRequest>
  <data>
    <Report>
      <name><![CDATA[with all parameters HTML_ZIPPED]]></name>
      <description><![CDATA[A scan report with ignored
       findings]]></description>
      <format>HTML_ZIPPED</format>
      <type>WAS_SCAN_REPORT</type>
      <config>
        <scanReport>
          <target>
            <scans>
              <WasScan>
```

```
                <id>104268</id>
              </WasScan>
            </scans>
          </target>
          <display>
            <contents>
              <ScanReportContent>DESCRIPTION</ScanReportContent>
              <ScanReportContent>SUMMARY</ScanReportContent>
              <ScanReportContent>GRAPHS</ScanReportContent>
              <ScanReportContent>RESULTS</ScanReportContent>
          <ScanReportContent>INDIVIDUAL_RECORDS</ScanReportContent>
              <ScanReportContent>RECORD_DETAILS</ScanReportContent>
              <ScanReportContent>ALL_RESULTS</ScanReportContent>
              <ScanReportContent>APPENDIX</ScanReportContent>
            </contents>
            <graphs>
              <ScanReportGraph>VULNERABILITIES_BY_SEVERITY</ScanReport
Graph>
              <ScanReportGraph>VULNERABILITIES_BY_GROUP</ScanReportGra
ph>
              <ScanReportGraph>VULNERABILITIES_BY_OWASP</ScanReportGra
ph>
              <ScanReportGraph>VULNERABILITIES_BY_WASC</ScanReportGrap
h>
              <ScanReportGraph>SENSITIVE_CONTENTS_BY_GROUP</ScanReport
Graph>
            </graphs>
            <groups>
              <ScanReportGroup>URL</ScanReportGroup>
              <ScanReportGroup>GROUP</ScanReportGroup>
              <ScanReportGroup>OWASP</ScanReportGroup>
              <ScanReportGroup>WASC</ScanReportGroup>
              <ScanReportGroup>STATUS</ScanReportGroup>
              <ScanReportGroup>CATEGORY</ScanReportGroup>
              <ScanReportGroup>QID</ScanReportGroup>
            </groups>
            <options>
              <rawLevels>true</rawLevels>
            </options>
          </display>
          <filters>
            <searchlists>
              <SearchList>
                <id>43147</id>
              </SearchList>
            </searchlists>
            <url>http://www.mysite.com/help.html</url>
            <remediation>
                <showIgnored>SHOW_BOTH</showIgnored>
                <ignoredReasons>
                  <IgnoredReason>FALSE_POSITIVE</IgnoredReason>
                  <IgnoredReason>RISK_ACCEPTED</IgnoredReason>
                  <IgnoredReason>NOT_APPLICABLE</IgnoredReason>
                </ignoredReasons>
            </remediation>
          </filters>
        </scanReport>
      </config>
    </Report>
  </data>
</ServiceRequest>
```

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/3.
0/was/report.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Report>
            <id>202447</id>
        </Report>
    </data>
</ServiceResponse>
```

Sample - Create a scan report using report template

Let’s generate a scan report in PDF format using a specific template (identified by its template ID).

API request

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <Report>
            <name><![CDATA[Scan Report for Servers]]></name>
                <format>PDF</format>
               <template>
                   <id>876049</id>
               </template>
                <config>
                    <scanReport>
                <target>
                <scans>
                <WasScan>
                <id>2252466</id>
                </WasScan>
                </scans>
                  </target>
                </scanReport>
            </config>
        </Report>
    </data>
</ServiceRequest>
```

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/3.
0/
was/report.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Report>
            <id>973057</id>
        </Report>
    </data>
</ServiceResponse>
```

Sample - Create a scan report in CSV_V2 format

Let’s generate a scan report in CSV-V2 format.

API request

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
   <data>
      <Report>
         <name><![CDATA[Scan Report for Servers]]></name>
         <format>CSV_V2</format>
         <template>
            <id>46441</id>
         </template>
         <config>
            <scanReport>
               <target>
                  <scans>
                     <WasScan>
                        <id>1667002</id>
                     </WasScan>
                  </scans>
               </target>
            </scanReport>
         </config>
      </Report>
   </data>
</ServiceRequest>
```

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/3.
0/was/report.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Report>
            <id>214159</id>
        </Report>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Scorecard Report

/qps/rest/3.0/create/was/report

[POST]

Using the Report Creation API you can create the Scorecard Report. A Scorecard Report ranks the vulnerability of your web applications.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and “Create Report”.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. For details, refer to Reference: Report Creation.

Click here for available operators

Parameter	Description
target.tags.included.option	(keyword: ALL or ANY) Decides which web applications should be included in the scan. ALL : Only the web applications associated with all the specified tags are included in the scan. ANY : Only the web applications associated with any of the specified tags included in the scan.
target.tags.included.tagList.Tag.id	(integer) The web applications associated with the tag (identified by the specified tag ID) are included in the scan.
filters.searchlists	(SearchList) Number of search lists to report on vulnerabilities in those lists. If

	no search lists are selected, the report will include all findings.
filters.scanDate	(DatetimeRange) Filter by Scan date.
filters.scanStatus	(WasScanConsolidatedStatus) Filter by scan status.
filters.scanAuthStatus	(WasScanAuthStatus) Filter by authentication status of the scan.
format	(keyword) Report format, one of: WORD, HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, CSV, CSV_V2, XML, POWERPOINT
display.contents	(ScorecardReportContent) DESCRIPTION, SUMMARY, GRAPHS, RES ULTS, INDIVIDUAL_RECORDS
target.tags.excluded.option	(keyword: ALL or ANY) Decides which web applications should be excluded from the scan. ALL : Only the web applications associated with all the specified tags are excluded from the scan. ANY : Only the web applications associated with any of the specified tags are excluded from the scan.
target.tags.excluded.tagList.Tag.id	(integer) The web applications associated with the tag (identified by the specified tag ID) are excluded from the scan.
display.graphs	(ScorecardReportGraph) The graphs to be included in the report: VULNERABILITIES_BY_SEVERITY,

	VULNERABILITIES_BY_GROUP, VULNERABILITIES_BY_OWASP, VULNERABILITIES_BY_WASC, SENSITIVE_CONTENTS_BY_GROUP, MOST_VULNERABLE_WEB_APPLICATIONS, OPERATING_SYSTEMS_DETECTED
display.groups	(ScorecardReportGroup) The group category to be included in the report: GROUP, OWASP, WASC.
display.options	(boolean) Display Options used/not used by the scorecard report.

Sample - Create a scorecard report

Let us create a scorecard report in PDF format, selecting a single tag for the target.

API request

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

Request POST data

```
<ServiceRequest>
  <data>
    <Report>
      <name><![CDATA[with all parameters PDF with rawLevel
false]]></name>
      <description><![CDATA[A simple scorecard report]]></description>
      <format>PDF</format>
      <type>WAS_SCORECARD_REPORT</type>
      <config>
        <scorecardReport>
          <target>
            <tags>
                 <included>
```

```
                   <option>ALL</option>
                     <tagList>
                      <Tag>
                       <id>7821676</id>
                      </Tag>
                     </tagList>
                 </included>
            </tags>
          </target>
          <display>
            <contents>
              <ScorecardReportContent>DESCRIPTION</ScorecardReportCont
ent>
              <ScorecardReportContent>SUMMARY</ScorecardReportContent>
              <ScorecardReportContent>GRAPHS</ScorecardReportContent>
              <ScorecardReportContent>RESULTS</ScorecardReportContent>
            </contents>
            <graphs>
         <ScorecardReportGraph>VULNERABILITIES_BY_GROUP</ScorecardRepo
rtGraph>
         <ScorecardReportGraph>VULNERABILITIES_BY_OWASP</ScorecardRepo
rtGraph>
         <ScorecardReportGraph>VULNERABILITIES_BY_WASC</ScorecardRepor
tGraph>
           </graphs>
           <groups>
             <ScorecardReportGroup>GROUP</ScorecardReportGroup>
             <ScorecardReportGroup>OWASP</ScorecardReportGroup>
             <ScorecardReportGroup>WASC</ScorecardReportGroup>
           </groups>
           <options>
             <rawLevels>false</rawLevels>
           </options>
            </display>
            <filters>
              <searchlists>
                <SearchList>
                  <id>43147</id>
                </SearchList>
                <SearchList>
                  <id>43147</id>
                </SearchList>
              </searchlists>
              <scanDate>
                <startDate>2017-08-28</startDate>
                <endDate>2017-10-28</endDate>
              </scanDate>
              <scanStatus>NO_HOST_ALIVE</scanStatus>
              <scanAuthStatus>NONE</scanAuthStatus>
            </filters>
          </scorecardReport>
        </config>
      </Report>
   </data>
</ServiceRequest>
```

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/3.
0/was/report.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Report>
      <id>4629</id>
    </Report>
  </data>
</ServiceResponse>
```

Sample - Create a scorecard report using the report template

Let’s generate a scorecard report in HTML format using a specific template (identified by its template ID).

API request

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

Request POST data

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

```
            <name>Report_08</name>
            <description><![CDATA[A scorecard report]]></description>
            <type>WAS_SCORECARD_REPORT</type>
            <format>HTML_ZIPPED</format>
            <template>
                <id>876051</id>
            </template>
            <config>
                <scorecardReport>
                    <target>
                        <tags>
                            <included>
                            <option>ALL</option>
                            <tagList>
                                <Tag>
                                    <id>11999629</id>
                                </Tag>
                                </tagList>
                            </included>
                            <excluded>
                            <option>ANY</option>
                            <tagList>
                                <Tag>
                                    <id>11999629</id>
                                </Tag>
                                </tagList>
                            </excluded>
                        </tags>
                    </target>
                </scorecardReport>
            </config>
        </Report>
    </data>
</ServiceRequest>
```

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/3.
0/
was/report.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
```

```
        <Report>
            <id>973058</id>
        </Report>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Catalog Report

/qps/rest/3.0/create/was/report

[POST]

Using the Report Creation API you can create the Catalog Report. A Catalog Report shows you the number and status of entries in your web application catalog.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and “Create Report”.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. The element “target” is required and at least one “scans” child element is required. For details, refer to Reference: Report Creation.

Click here for available operators

Parameter	Description
filters.scanDate	(DatetimeRange) Filter by scan date.
filters.url	(text) Filter by web app URL.
filters.ip	(text) Filter by IP address.
filters.os	(text) Filter by OS.
filters.status	(EntryStatus) Filter by status.
format	(keyword) Report format, one of: WORD, HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, CSV, CSV_V2, XML, POWERPOINT

display.contents	(CatalogReportContent) The report content: Description, Summary, Results, Individual Records, Graphs.
display.graphs	(CatalogReportGraph) The graphs to be included in the report: ENTRIES_BY_STATUS, ENTRIES_ADDED_OVER_TIME, OPERATING_SYSTEMS_DETECTED.
display.groups	(CatalogReportGroup) The group category to be included in the report: STATUS, OPERATING_SYSTEM.

Sample - Create a catalog report

Let us create a catalog report in CSV format, selecting a single tag for the target.

API request

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

Request POST data

```
<ServiceRequest>
  <data>
    <Report>
      <name><![CDATA[with all parameters CSV]]></name>
      <description><![CDATA[A simple Catalog report]]></description>
      <type>WAS_CATALOG_REPORT</type>
      <format>CSV</format>
      <config>
        <catalogReport>
          <display>
            <contents>
              <CatalogReportContent>DESCRIPTION</CatalogReportContent>
              <CatalogReportContent>SUMMARY</CatalogReportContent>
              <CatalogReportContent>GRAPHS</CatalogReportContent>
              <CatalogReportContent>RESULTS</CatalogReportContent>
              <CatalogReportContent>INDIVIDUAL_RECORDS</CatalogReportC
ontent>
            </contents>
            <graphs>
              <CatalogReportGraph>ENTRIES_ADDED_OVER_TIME</CatalogRepo
rtGraph>
              <CatalogReportGraph>ENTRIES_BY_STATUS</CatalogReportGrap
h>
            </graphs>
            <groups>
                <CatalogReportGroup>STATUS</CatalogReportGroup>
              <CatalogReportGroup>OPERATING_SYSTEM</CatalogReportGroup
>
            </groups>
          </display>
          <filters>
            <status>
               <EntryStatus>NEW</EntryStatus>
              <EntryStatus>SUBSCRIPTION</EntryStatus>
              <EntryStatus>ROGUE</EntryStatus>
              <EntryStatus>APPROVED</EntryStatus>
              <EntryStatus>REJECTED</EntryStatus>
            </status>
            <scanDate>
              <startDate>2017-06-29</startDate>
              <endDate>2017-06-29</endDate>
            </scanDate>
            <url><![CDATA[mysite.fr]]></url>
            <os><![CDATA[unix]]></os>
          </filters>
        </catalogReport>
      </config>
    </Report>
  </data>
</ServiceRequest>
```

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/3.
0/was/report.xsd">
<responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
```

```
    <Report>
      <id>5629</id>
    </Report>
  </data>
</ServiceResponse>
```

Sample - Create a catalog report using report template

Let’s generate a catalog report in PDF format using a specific template (identified by its template ID).

API request

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
    <data>
        <Report>
            <name><![CDATA[Catalog Report for Servers]]></name>
            <description><![CDATA[A simple catalog
report]]></description>
            <format>PDF</format>
             <template>
                   <id>876050</id>
            </template>
        </Report>
    </data>
</ServiceRequest>
```

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/3.0/was/report.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
```

```
        <Report>
            <id>973058</id>
        </Report>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Report Template Count

/qps/rest/3.0/count/was/reporttemplate

[POST]

Returns the total number of report templates in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access".

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. See Reference: Report Creation for details.

Click here for available operators

Parameter	Description
id	(integer) The report ID. This element is assigned by the service and is required for a certain type of request (details, status, update, delete, send or download).
name	(text) A report name (maximum 256 characters). Applies to all reports.
type	(keyword) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT

Sample - Count the report templates

You can search for templates by using different filters for template ID, template name or type of report. Let's consider an example of searching report template using filter for template ID.

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="EQUALS">1234</Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/
was/reporttemplate.xsd">
      <responseCode>SUCCESS</responseCode>
            <count>6</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Search Report Template

/qps/rest/3.0/search/was/reporttemplate

[POST]

You can search for existing report templates

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. The element “target” is required and at least one “scans” child element is required. See Reference: Report Creation for details.

Click here for available operators

Parameter	Description
id	(integer) The report ID. This element is assigned by the service and is required for a certain type of request (details, status, update, delete, send or download).
name	(text) A report name (maximum 256 characters). Applies to all reports.
type	(keyword) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT

Sample - Search report templates

You can search for templates by using different filters for template ID, template name or type of report. Let’s consider an example of searching report template using filter for template ID.

API request

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

Request POST data

```
<ServiceRequest>
   <filters>
      <Criteria field="id" operator="EQUALS">876048</Criteria>
   </filters>
</ServiceRequest>
```

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/3.
0/
was/reporttemplate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <ReportTemplate>
            <id>876048</id>
            <name><![CDATA[Web Application Report]]></name>
            <description>
                <![CDATA[Each targeted web application is listed with
the total number of detected vulnerabilities and sensitive content.]]>
            </description>
            <owner>
                <id>23220145</id>
                <username>username</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Smith]]></lastName>
            </owner>
            <type>WAS_WEBAPP_REPORT</type>
            <creationDate>2017-04-11T09:29:23Z</creationDate>
            <tags>
                <count>0</count>
            </tags>
            <config>
                <webAppReportTemplate>
```

```
                    <display>
                        <contents>
<WebAppReportContent>DESCRIPTION</WebAppReportContent>
<WebAppReportContent>SUMMARY</WebAppReportContent>
<WebAppReportContent>GRAPHS</WebAppReportContent>
<WebAppReportContent>RESULTS</WebAppReportContent>
<WebAppReportContent>INDIVIDUAL_RECORDS</WebAppReportContent>
<WebAppReportContent>RECORD_DETAILS</WebAppReportContent>
<WebAppReportContent>APPENDIX</WebAppReportContent>
                        </contents>
                        <graphs>
<WebAppReportGraph>VULNERABILITIES_BY_SEVERITY</WebAppReportGraph>
<WebAppReportGraph>VULNERABILITIES_BY_STATUS</WebAppReportGraph>
<WebAppReportGraph>VULNERABILITIES_BY_GROUP</WebAppReportGraph>
<WebAppReportGraph>VULNERABILITIES_BY_OWASP</WebAppReportGraph>
                        </graphs>
                        <groups>
                          <WebAppReportGroup>WEBAPP</WebAppReportGroup
>
                          <WebAppReportGroup>CATEGORY</WebAppReportGro
up>
                          <WebAppReportGroup>GROUP</WebAppReportGroup>
                          <WebAppReportGroup>QID</WebAppReportGroup>
                        </groups>
                        <options>
     <rawLevels>true</rawLevels>
                        </options>
                    </display>
                    <filters>
                        <includedSearchLists/>
                        <excludedSearchLists/>
                        <url><![CDATA[null]]></url>
                        <status>
                         <WebAppFindingStatus>NEW</WebAppFindingStatus
>
                           <WebAppFindingStatus>ACTIVE</WebAppFindingS
tatus>
                       <WebAppFindingStatus>REOPENED</WebAppFindingSta
tus>
                        </status>
                        <remediation>
                            <showPatched>SHOW_BOTH</showPatched>
                            <showIgnored>SHOW_NONE</showIgnored>
                            <ignoredReasons>
<IgnoredReason>NOT_APPLICABLE</IgnoredReason>
```

```
<IgnoredReason>FALSE_POSITIVE</IgnoredReason>
                                <IgnoredReason>RISK_ACCEPTED</IgnoredR
eason>
                            </ignoredReasons>
                        </remediation>
                    </filters>
                </webAppReportTemplate>
            </config>
        </ReportTemplate>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Get details of Report Template

/qps/rest/3.0/get/was/reporttemplate/<id>

[GET]

View details for a report template which is in the user’s scope. See “Search Report Template” to find a record ID to use as input.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”.

Input Parameters

The element “id” (integer) is required, where “id” identifies the report.

Click here for available operators

Sample - Get details of the report template

Let us get details of a report template.

API request

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/reporttemplate/8760
48"
```

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/3.
0/was/reporttemplate.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <ReportTemplate>
            <id>876048</id>
            <name><![CDATA[Web Application Report]]></name>
            <description>
```

```
                <![CDATA[Each targeted web application is listed with
the total number of detected vulnerabilities and sensitive content.]]>
            </description>
            <owner>
                <id>23220145</id>
                <username>john_doe</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <type>WAS_WEBAPP_REPORT</type>
            <creationDate>2017-04-11T09:29:23Z</creationDate>
            <tags>
                <count>0</count>
            </tags>
            <config>
                <webAppReportTemplate>
                    <display>
                        <contents>
<WebAppReportContent>DESCRIPTION</WebAppReportContent>
           <WebAppReportContent>SUMMARY</WebAppReportContent>
                  <WebAppReportContent>GRAPHS</WebAppReportContent>
                         <WebAppReportContent>RESULTS</WebAppReportCon
tent>                            <WebAppReportContent>INDIVIDUAL_RECOR
DS</WebAppReportContent>                            <WebAppReportConte
nt>RECORD_DETAILS</WebAppReportContent>                            <We
bAppReportContent>APPENDIX</WebAppReportContent>
  </contents>
                        <graphs>
<WebAppReportGraph>VULNERABILITIES_BY_SEVERITY</WebAppReportGraph>
                        <WebAppReportGraph>VULNERABILITIES_BY_STATUS</
WebAppReportGraph>                            <WebAppReportGraph>VULNE
RABILITIES_BY_GROUP</WebAppReportGraph>                            <We
bAppReportGraph>VULNERABILITIES_BY_OWASP</WebAppReportGraph>
              </graphs>
                        <groups>
                            <WebAppReportGroup>WEBAPP</WebAppReportGro
up>
                            <WebAppReportGroup>CATEGORY</WebAppReportG
roup>
                            <WebAppReportGroup>GROUP</WebAppReportGrou
p>
                            <WebAppReportGroup>QID</WebAppReportGroup>
                        </groups>
                        <options>
                            <rawLevels>true</rawLevels>
                        </options>
                    </display>
                    <filters>
                        <includedSearchLists/>
                        <excludedSearchLists/>
                        <url><![CDATA[null]]></url>
                        <status>
                           <WebAppFindingStatus>NEW</WebAppFindingStat
us>
                           <WebAppFindingStatus>ACTIVE</WebAppFindingS
tatus>
                        <WebAppFindingStatus>REOPENED</WebAppFindingSt
atus>
                        </status>
                        <remediation>
                            <showPatched>SHOW_BOTH</showPatched>
                            <showIgnored>SHOW_NONE</showIgnored>
                            <ignoredReasons>
                               <IgnoredReason>NOT_APPLICABLE</IgnoredR
eason>
<IgnoredReason>FALSE_POSITIVE</IgnoredReason>
                                <IgnoredReason>RISK_ACCEPTED</IgnoredR
eason>
                            </ignoredReasons>
                        </remediation>
                    </filters>
                </webAppReportTemplate>
            </config>
        </ReportTemplate>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/report.xsd

Reference: Report

The <Report> element includes sub elements used to define a web application report. A reference of these elements is provided below. An asterisk * indicates a complex element.

Parameter	Description
id	(integer) The report ID. This element is assigned by the service and is required for a certain type of request (details, status, update, delete, send or download).
name	(text) A report name (maximum 256 characters). Applies to all reports.
tags.id	(integer) ID of the tag associated with the report.
tags.name	(text) Name of the tag associated with the report.
creationDate	(date) The date when the report was created in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).
type	(keyword) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT, DATALIST_REPORT.
format	(keyword) The format of the report, one of: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, POWERPOINT, CSV, CSV_V2, XML, WORD.
status	(keyword) The status of the report: RUNNING, ERROR or COMPLETE.

Parameter	Description
id	(integer) The report ID. This element is assigned by the service and is required for a certain type of request (details, status, update, delete, send or download).

name

(text) A report name (maximum 256 characters). Applies to all reports.

Note: Generating a report without template will allow you to assign a name to the report. If you use template during report generation, the name you provide in the request is ignored and the template name is assigned to the report.

description

(text) A description of the report.

owner*

This element is assigned by the service and may be specified for an update request only.

Example:

```
<owner>
  <id>123056</id>
  <username>username</username>
  <firstName><![CDATA[Johns]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</owner>
```

type

(text) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT, DATALIST_REPORT

format

(text) The format of the report, one of: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, POWERPOINT, CSV, CSV_V2, XML, WORD

tags*

This element identifies the tags associated with the report.

```
Example:
<tags>
  <count>2</count>
  <list>
    <Tag>
      <id>99509</id>
      <name><![CDATA[Tag 1]]></name>
   </Tag>
   <Tag>
     <id>99511</id>
     <name><![CDATA[Tag 2]]></name>
    </Tag>
  </list>
</tags>
```

password

(text) The password for a PDF encrypted report.

distributionList*

This element specifies the email addresses for distribution of the report.

```
Example:
<distributionList>
     <set>
          <EmailAddress><![CDATA[abc1@qualys.com]]></
EmailAddress>
          <EmailAddress><![CDATA[abc2@qualys.com]]></
EmailAddress>
      </set>
</distributionList>
```

config*

...

The configuration options for report creation.

```
Example:
<config>
  <webAppReport>
     <target>
       <tags>
         <Tag>
           <id>102609</id>
         </Tag>
       </tags>
       <webapps>
         <WebApp>
            <id>324538</id>
          </WebApp>
        </webapps>
      </target>
```

status

(keyword) The status of the report: RUNNING, ERROR or COMPLETE

creationDate

(date) The date when the report was created in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).

lastDownloadDate	(date) The date when the report was last downloaded in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).
downloadCount	(integer) The number of times the report has been downloaded.

Reference: Report Creation

The Report “config” element includes sub elements used to define a web application report type. A reference of these elements is provided below. An asterisk * indicates a complex element.

Parameter	Description
id	(integer) The report ID. This element is assigned by the service and is required for a certain type of request (details, status, update, delete, send or download).
name	(text) A report name (maximum 256 characters). Applies to all reports. Note: Generating a report without template will allow you to assign a name to the report. If you use template during report generation, the name you provide in the request is ignored and the template name is assigned to the report.

target*	A report target. Applies to all reports.
	Example for a web application report:
	`<tags>`
	`     <included>`
	`       <option>ALL</option>`
	`         <tagList>`
	`           <set>`
	`             <Tag><id>12017424</id></Tag>`
	`             <Tag><id>12017228</id></Tag>`
	`           </set>`
	`         </tagList>`
	`         </included>`
	`         <excluded>`
	`            <option>ANY</option>`
	`                <tagList>`
	`                   <set>`
	`                     <Tag><id>12017228</id></Tag>`
	`                 </set>`
	`                </tagList>`

	`</excluded>`
	`</tags>`

template.id	(integer) The template ID. This element is assigned by the system and is required
	for a certain type of request.
	Example:
	`<template>`
	`<id>876048</id>`
	`</template>`

type (text)	The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT, DATALIST_REPORT
password (text)	A password for a encrypted PDF report. Applies to all reports.

distributionList*	Email addresses for a report distribution list. Applies to all reports.
	Example:
	`<distributionList>`
	`<set>`
	`<EmailAddress><![CDATA[abc1@qualys.com]]`
	`></EmailAddress>`
	`<EmailAddress><![CDATA[abc2@qualys.com]]`
	`></EmailAddress>`
	`</set>`
	`</distributionList>`

display.contents*	Identifies the report content to display. Values: DESCRIPTION, SUMMARY, GRAPHS, RESULTS, INDIVIDUAL_RECORDS (all reports)

Values: RECORD_DETAILS, ALL_RESULTS, APPENDIX (Web Application Report and Scan Report)

Example for a Scan Report:

```
<display>
 <contents>
  <ScanReportContent>GRAPHS</ScanReportContent>
  <ScanReportContent>RESULTS</ScanReportContent>
 </contents>
</display>
```

display.graphs*

Identifies the graphs to display. Applies to all reports.

Example for a Scan Report:

```
<display>
<graphs>
    <ScanReportGraph>
         MOST_VULNERABLE_URLS
   </ScanReportGraph>
    <ScanReportGraph>
          VULNERABILITIES_BY_SEVERITY
    </ScanReportGraph>
    <ScanReportGraph>
          VULNERABILITIES_BY_GROUP
   </ScanReportGraph>
    <ScanReportGraph>
          VULNERABILITIES_BY_OWASP
    </ScanReportGraph>
    <ScanReportGraph>
          VULNERABILITIES_BY_WASC
    </ScanReportGraph>
    <ScanReportGraph>
          SENSITIVE_CONTENTS_BY_GROUP
    </ScanReportGraph>
 </graphs>
</display>
```

display.groups*

Identifies the vulnerability groups to display. Applies to all reports.

Example for a Web Application Report or Scan Report:

```
<display>
  <groups>
    <WebAppReportGroup>GROUP</WebAppReportGroup>
    <WebAppReportGroup>OWASP</WebAppReportGroup>
    <WebAppReportGroup>WASC</WebAppReportGroup>
  </groups>
</display>
```

display.options*

Specifies whether to display severity using levels (1 through 5) or using ratings (low, medium, high). Applies to all reports.

filters.searchlists*

Identifies search list filters. Applies to a Web Application Report, Scan Report or Scorecard Report.

```
Example:
<filters>
  <SearchLists>
    <SearchList>
      <id>43147</id>
    </SearchList>
  </SearchlLsts>
  </filters>
```

filters.url (text)

Identifies URL filters. Applies to a Web Application Report, Scan Report or Catalog Report.

```
Example:
<filters>
<url>http://www.mysite.com/help.html</url>
...
</filters>
```

filters.status*

Identifies status filters. Applies to Web Application Report, Scan Report and Catalog Report.

Values for Web Application Report and Scan Report: NEW, ACTIVE, REOPENED, FIXED

Values for Catalog Report: NEW, ROGUE, APPROVED, REJECTED, SUBSCRIPTION

filters.showPatched (keyword)	Identifies whether to include/not include findings with virtual patches. Applies to Web Application Report and Scan Report. Values: SHOW_ONLY - show patched findings only SHOW_BOTH - show patched & unpatched findings (default) SHOW_NONE - show unpatched findings only
filters.remediation. showIgnored (boolean)	Include ignored findings: true or false
filters.remediation. ignoredReasons (keyword)	Identifies the types of findings to be included in the report.Applies to Scan Report. Values: FALSE_POSITIVE - include false positive findings in the report RISK_ACCEPTED - include risk accepted findings in the report NOT_APPLICABLE - include findings marked as not applicable in the report

filters.scanDate*	Applies to a Scorecard Report and Catalog Report.
	`Example:`
	`<filters>`
	` <scanDate>`
	`   <startDate>2017-08-28</startDate>`
	`   <endDate>2017-10-28</endDate>`
	`  </scanDate>`
	`</filters>`

filters.scanStatus *	Applies to a Scorecard Report. Tip - Specify SERVICE_ERROR to include scans with the status Service Errors Detected.
	`Example:`
	`<filters>`
	` <scanStatus>FINISHED</scanStatus>`
	`</filters>`

filters.scanAuthStatus*	Applies to a Scorecard Report
	`Example:`
	`<filters>`
	` <scanAuthStatus>SUCCESSFUL</scanAuthStatus>`
	`</filters>`

filters.ip (text)	Applies to a Catalog Report
	`Example:`
	`<filters>`
	` <ip><![CDATA[10.56.64.245]]></ip>`
	`</filters>`

filters.os (text)	Applies to a Catalog Report
	`Example:`
	`<filters>`
	` <os><![CDATA[unix]]></os>`
	`</filters>`

Findings

Finding Count

/qps/rest/3.0/count/was/finding

[POST]

Returns the total number of findings on web application(s) in the user’s scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”. The count includes web applications in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) ID of the finding (WebAppVuln, WebAppIg, or WebAppSensitiveContent).

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

qid	(integer) Qualys ID assigned to the detection.
name	(text) Name of the detection finding.
type	(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.
url	(text) URL of the web application on which the finding was detected.
webApp.tags.id	(date) ID of the tag associated with the web application on which the finding was detected.
webApp.tags.name	(text) Name of the tag associated with the web application on which the finding was detected.
status	(keyword) Status of the finding: NEW, ACTIVE, REOPENED, PROTECTED and FIXED.
patch	(integer-long) Use WAF to protect against vulnerabilities by installing virtual patches.
webApp.id	(integer) ID of the web application on which the finding was detected.
webApp.name	(text) Name of the web application on which the finding was detected.
severity	(integer) Severity of the finding.
externalRef	(string) Tip - Use operator IS EMPTY for findings with empty external references.
ignoredDate	(date) The date on which the finding was marked to ignore.
ignoredReason	(keyword) The reason for which the finding is ignored: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE

group	(keyword) XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM
owasp.name	(text) Name of the OWASP vulnerability.
owasp.code	(integer) Code associated with the OWASP vulnerability
wasc.name	(text) Name of the vulnerability.
wasc.code	(integer) Code of the vulnerability.
cwe.id	(integer) ID associated with CWE.
firstDetectedDate	(date) The date when the finding was first detected in the web application,
lastDetectedDate	(date) The date when the finding was last detected in the web application.
lastTestedDate	(date) The date when the finding was last tested in the web application.
timesDetected	(integer) The count indicating the number of times the finding was detected.
severity level	(integer) The severity associated with the finding:1,2,3,4,5

Sample - Get count of all findings

Return the number (count) of all findings in the user's scope.

API request

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

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/3.0
/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
     <count>2815</count>
</ServiceResponse>
```

Sample - Get count of findings with a criteria

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
<Criteria field="type"
operator="EQUALS">VULNERABILITY</Criteria>
<Criteria field="severity" operator="EQUALS">5</Criteria>
<Criteria field="status" operator="IN">NEW, ACTIVE,
REOPENED</Criteria>
</filters>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/3.0
/was/finding.xsd">
      <responseCode>SUCCESS</responseCode>
         <count>41</count>
</ServiceResponse>
```

Sample - Get details of finding

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

API request

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

Request POST data

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

XML 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/3.0
/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/finding.xsd

Search Findings

/qps/rest/3.0/search/was/finding

[POST]

Returns list of findings (vulnerabilities, sensitive contents, information gathered) found in web applications which are in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The output includes findings in the user's scope.

We have added the <fixedDate> XML tag to the Search Findings API. With this enhancement you can search findings based on vulnerability fixed date.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) ID of the finding (WebAppVuln, WebAppIg, or WebAppSensitiveContent).

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

qid	(integer) Qualys ID assigned to the detection.

name	(text) Name of the detection finding.
type	(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.
url	(text) URL of the web application on which the finding was detected.
webApp.tags.id	(date) ID of the tag associated with the web application on which the finding was detected.
webApp.tags.name	(text) Name of the tag associated with the web application on which the finding was detected.
status	(keyword) Status of the finding: NEW, ACTIVE, REOPENED, PROTECTED, and FIXED.
patch	(integer-long) Use WAF to protect against vulnerabilities by installing virtual patches.
webApp.id	(integer) ID of the web application on which the finding was detected.
webApp.name	(text) Name of the web application on which the finding was detected.
severity	(integer) Severity of the finding.
externalRef	(string) Tip - Use operator IS EMPTY for findings with empty external references.
ignoredDate	(date) The date on which the finding was marked to ignore.
ignoredReason	(keyword) The reason for which the finding is ignored: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE

group	(keyword) XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM
owasp.name	(text) Name of the OWASP vulnerability.
owasp.code	(integer) Code associated with the OWASP vulnerability
wasc.name	(text) Name of the vulnerability.
wasc.code	(integer) Code of the vulnerability.
cwe.id	(integer) ID associated with CWE.
firstDetectedDate	(date) The date when the finding was first detected in the web application,
lastDetectedDate	(date) The date when the finding was last detected in the web application.
lastTestedDate	(date) The date when the finding was last tested in the web application.
timesDetected	(integer) The count indicating the number of times the finding was detected.
severity level	(integer) The severity associated with the finding:1,2,3,4,5
fixedDate	(date) The vulnerability fixed date for findings.

Sample - Search for finding with specific ID

API request

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

Request POST data

```
<ServiceRequest>
  <preferences>
    <verbose>true</verbose>
  </preferences>
  <filters>
    <Criteria field="id" operator="EQUALS">156582</Criteria>
  </filters>
</ServiceRequest>
```

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/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <Finding>
            <id>156582</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
            <qid>150124</qid>
            <name>
                <![CDATA[Clickjacking - Framable Page]]>
            </name>
            <type>VULNERABILITY</type>
            <findingType>QUALYS</findingType>
            <cwe>
                <count>1</count>
                <list>
                    <long>451</long>
                </list>
            </cwe>
            <owasp>
                <count>1</count>
                <list>
                    <OWASP>
                        <name>
                            <![CDATA[Security Misconfiguration]]>
                        </name>
                        <url>
                            <![CDATA[https://www.owasp.org/index.php/T
op_10-2017_A6-Security_Misconfiguration]]>
```

```
                        </url>
                        <code>6</code>
                    </OWASP>
                </list>
            </owasp>
            <wasc>
                <count>1</count>
                <list>
                    <WASC>
                        <name>
                            <![CDATA[Application Misconfiguration]]>
                        </name>
                        <url>
                            <![CDATA[http://projects.webappsec.org/w/p
age/13246914/WASC]]>
                        </url>
                        <code>15</code>
                    </WASC>
                </list>
            </wasc>
            <resultList>
                <count>1</count>
                <list>
                    <Result>
                        <authentication>false</authentication>
                        <ajax>false</ajax>
                        <payloads>
                            <count>1</count>
                            <list>
                                <PayloadInstance>
                                    <payload>
                                        <![CDATA[N/A]]>
                                    </payload>
                                    <request>
                                        <method>
                                            <![CDATA[GET]]>
                                        </method>
                                        <link>
                                            <![CDATA[http://funkytown.
vuln.qa.qualys.com/cassium/xss/]]>
                                        </link>
                                        <headers>
                                            <![CDATA[]]>
                                        </headers>
                                    </request>
                                    <response>
                                        <![CDATA[The URI was framed.
]]>
                                    </response>
                                </PayloadInstance>
                            </list>
                        </payloads>
                    </Result>
                </list>
            </resultList>
            <severity>3</severity>
            <url>
                <![CDATA[http://funkytown.vuln.qa.qualys.com/cassium/x
ss/]]>
            </url>
            <status>ACTIVE</status>
            <firstDetectedDate>2017-04-
28T09:36:13Z</firstDetectedDate>
            <lastDetectedDate>2018-02-21T09:03:32Z</lastDetectedDate>
            <lastTestedDate>2018-02-21T09:03:32Z</lastTestedDate>
            <timesDetected>3</timesDetected>
            <webApp>
                <id>286824</id>
                <name>
                    <![CDATA[webapp]]>
                </name>
                <url>
                    <![CDATA[http://funkytown.vuln.qa.qualys.com:80/ca
ssium/xss/]]>
                </url>
                <tags>
                  <count>2</count>
                  <list>
                    <Tag>
                        <id>8753812</id>
                        <name>
                            <![CDATA[Multiscan]]>
                        </name>
                    </Tag>
                    <Tag>
                        <id>9029017</id>
                        <name>
                            <![CDATA[TagWebapp1]]>
                        </name>
                    </Tag>
                </list>
            </tags>
        </webApp>
        <isIgnored>true</isIgnored>
          <ignoredReason>FALSE_POSITIVE</ignoredReason>
             <ignoredBy>
                <id>1056860</id>
                <username>user_john</username>
                <firstName>
                        <![CDATA[John]]>
                </firstName>
                <lastName>
                        <![CDATA[Doe]]>
                </lastName>
             </ignoredBy>
             <ignoredDate>2019-03-04T03:19:29Z</ignoredDate>
             <ignoredComment>
                    <![CDATA[This is test comment]]>
             </ignoredComment>
            <retest/>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - Search with criteria: condensed response

API request

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

Request POST data

```
<ServiceRequest>
 <filters>
   <Criteria field="id" operator="EQUALS">935943</Criteria>
 </filters>
</ServiceRequest>
```

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/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <Finding>
            <id>935943</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
            <qid>150117</qid>
            <name>
                <![CDATA[Path-Based Cross-Site Scripting (XSS)]]>
            </name>
            <type>VULNERABILITY</type>
            <findingType>QUALYS</findingType>
            <severity>5</severity>
            <url>
                <![CDATA[http://funkytown.vuln.qa.example.com/cassium/
traversal/page_48/%22%3e%3cimg%20src%3dq%20onerror%3dalert(9)%3e]]>
            </url>
            <status>ACTIVE</status>
            <firstDetectedDate>2017-04-
04T06:15:33Z</firstDetectedDate>
            <lastDetectedDate>2017-04-04T06:16:20Z</lastDetectedDate>
            <lastTestedDate>2017-04-04T06:16:20Z</lastTestedDate>
            <timesDetected>3</timesDetected>
            <webApp>
                <id>4080112</id>
                <name>
                    <![CDATA[web app 1491286489688]]>
                </name>
                <url>
                    <![CDATA[http://funkytown.vuln.qa.example.com:80/c
assium/xss/]]>
                </url>
            </webApp>
            <isIgnored>true</isIgnored>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - Search with criteria: condensed response

API request

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

Request POST data

```
<ServiceRequest>
 <filters>
   <Criteria field="id" operator="EQUALS">935943</Criteria>
 </filters>
</ServiceRequest>
```

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/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <Finding>
            <id>935943</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
            <qid>150117</qid>
            <name>
                <![CDATA[Path-Based Cross-Site Scripting (XSS)]]>
            </name>
            <type>VULNERABILITY</type>
            <findingType>QUALYS</findingType>
            <severity>5</severity>
            <url>
                <![CDATA[http://funkytown.vuln.qa.example.com/cassium/
traversal/page_48/%22%3e%3cimg%20src%3dq%20onerror%3dalert(9)%3e]]>
            </url>
            <status>ACTIVE</status>
            <firstDetectedDate>2017-04-
04T06:15:33Z</firstDetectedDate>
            <lastDetectedDate>2017-04-04T06:16:20Z</lastDetectedDate>
            <lastTestedDate>2017-04-04T06:16:20Z</lastTestedDate>
            <timesDetected>3</timesDetected>
            <webApp>
                <id>4080112</id>
                <name>
                    <![CDATA[web app 1491286489688]]>
                </name>
                <url>
                    <![CDATA[http://funkytown.vuln.qa.example.com:80/c
assium/xss/]]>
                </url>
            </webApp>
            <isIgnored>true</isIgnored>
        </Finding>
    </data>
```

Sample - Search finding using uniqueId

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

API request

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

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

Request POST data

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

XML 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/3.0
/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <Finding>
            <id>132990</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
            <qid>150004</qid>
            <name>
                <![CDATA[Path-Based Vulnerability]]>
            </name>
            <type>VULNERABILITY</type>
            <findingType>QUALYS</findingType>
            <cwe>
                <count>1</count>
                <list>
                    <long>22</long>
                </list>
            </cwe>
            ...
        </webApp>
        <isIgnored>false</isIgnored>
        <retest/>
    </Finding>
</data>
</ServiceResponse>
```

Sample - Search Findings using Vulnerability Fixed Date

Use the fixedDate parameter, to search findings based on vulnerability fixed date.

API request

```
curl -n -u "USERNAME:PASSWORD"
"<qualys_base_url>/qps/rest/3.0/search/was/finding/"
```

Request POST data

```
<ServiceRequest>
    <filters>
        <Criteria field="fixedDate" operator="EQUALS">2024-04-
08</Criteria>
    </filters>
</ServiceRequest>
```

XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="
    <qualys_base_url>/qps/xsd/3.0/was/finding.
xsd">
        <responseCode>SUCCESS</responseCode>
        <count>100</count>
        <hasMoreRecords>false</hasMoreRecords>
        <data>
            <Finding>
                <id>5550410</id>
                <uniqueId>1ccfbdab-f5fa-464e-b0e7-
e502dcc4cccd</uniqueId>
                <qid>150023</qid>
                <detectionScore>50</detectionScore>
                <name>
                    <![CDATA[Directory Listing]]>
                </name>
                <type>VULNERABILITY</type>
                <potential>false</potential>
                <findingType>QUALYS</findingType>
                <severity>2</severity>
                <url>
                    <![CDATA[https://10.11.68.26/Frameworks/]]>
                </url>
                <status>FIXED</status>
                <firstDetectedDate>2024-02-
08T10:49:48Z</firstDetectedDate>
                <lastDetectedDate>2024-02-
08T10:49:48Z</lastDetectedDate>
                <lastTestedDate>2024-02-08T18:37:08Z</lastTestedDate>
                <fixedDate>2024-04-08T18:37:08Z</fixedDate>
                <timesDetected>1</timesDetected>
                <webApp>
                    <id>20018939</id>
                    <name>
                        <![CDATA[Edited Name-SRD-Test Progression Ravi
New Web
App12]]>
                    </name>
                    <url>
                        <![CDATA[http://10.11.68.26]]>
                    </url>
                </webApp>
                <isIgnored>false</isIgnored>
            </Finding>

.........
.........
.........

            <Finding>
                <id>5550460</id>
                <uniqueId>6736ddce-f165-4146-a264-
a504799c1438</uniqueId>
                <qid>150023</qid>
                <detectionScore>50</detectionScore>
                <name>
                    <![CDATA[Directory Listing]]>
                </name>
                <type>VULNERABILITY</type>
                <potential>false</potential>
                <findingType>QUALYS</findingType>
                <severity>2</severity>
                <url>
                    <![CDATA[http://10.11.68.26/tmpbanamex/]]>
                </url>
                <status>FIXED</status>
                <firstDetectedDate>2024-02-
08T10:49:48Z</firstDetectedDate>
                <lastDetectedDate>2024-02-
08T10:49:48Z</lastDetectedDate>
                <lastTestedDate>2024-02-08T18:37:08Z</lastTestedDate>
                <fixedDate>2024-04-08T18:37:08Z</fixedDate>
                <timesDetected>1</timesDetected>
                <webApp>
                    <id>20018939</id>
                    <name>
                        <![CDATA[Edited Name-SRD-Test Progression Ravi
New Web
App12]]>
                    </name>
                    <url>
                        <![CDATA[http://10.11.68.26]]>
                    </url>
                </webApp>
                <isIgnored>false</isIgnored>
            </Finding>
```

```
........
........
........

            <Finding>
                <id>5550648</id>
                <uniqueId>868b9f75-ccb2-43ea-9226-
01a43fde7aab</uniqueId>
                <qid>150023</qid>
                <detectionScore>50</detectionScore>
                <name>
                    <![CDATA[Directory Listing]]>
                </name>
                <type>VULNERABILITY</type>
                <potential>false</potential>
                <findingType>QUALYS</findingType>
                <severity>2</severity>
                <url>
                    <![CDATA[http://10.11.68.26/SWAGGERAPI/CRM51703Pat
hFuzzingRulesRogersMethodLevel/]]>
                </url>
                <status>FIXED</status>
                <firstDetectedDate>2024-02-
08T10:49:48Z</firstDetectedDate>
                <lastDetectedDate>2024-02-
08T10:49:48Z</lastDetectedDate>
                <lastTestedDate>2024-02-08T18:37:08Z</lastTestedDate>
                <fixedDate>2024-02-08T18:37:08Z</fixedDate>
                <timesDetected>1</timesDetected>
                <webApp>
                    <id>20018939</id>
                    <name>
                        <![CDATA[Edited Name-SRD-Test Progression Ravi
New Web
App12]]>
                    </name>
                    <url>
                        <![CDATA[http://10.11.68.26]]>
                    </url>
                </webApp>
                <isIgnored>false</isIgnored>
            </Finding>
        </data>
    </ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/finding.xsd

Get Finding Details

/qps/rest/3.0/get/was/finding/<id>

[GET]

Returns details for a finding on a web application which is in the user’s scope. See “Search findings” to find a record ID to use as input? See Search Findings.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”. The output includes findings for web applications in the user's scope.

We added the <fixedDate> XML tag to the Get Finding Details API. With this enhancement you can see the vulnerability fixed date for a finding.

Input Parameters

The element “id” (integer) is required, where “id” identifies a finding (WebAppVuln, WebAppIg, or WebAppSensitiveContent).

Click here for available operators

Sample - View details for the finding

Let us view details for the web application with the ID 1729432.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/finding/1729432"
```

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/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
```

```
            <id>1729432</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
            <qid>150117</qid>
            <name>
                <![CDATA[Path-Based Cross-Site Scripting (XSS)]]>
            </name>
            <type>VULNERABILITY</type>
            <findingType>QUALYS</findingType>
            <group>XSS</group>
            <cwe>
                <count>1</count>
                <list>
                    <long>79</long>
                </list>
            </cwe>
            <owasp>
                <count>1</count>
                <list>
                    <OWASP>
                        <name>
                            <![CDATA[Cross-Site Scripting (XSS)]]>
                        </name>
                        <url>
                            <![CDATA[https://www.owasp.org/index.php/T
op_10-2017_A7-Cross-Site_Scripting_(XSS)]]>
                        </url>
                        <code>7</code>
                    </OWASP>
                </list>
            </owasp>
            <wasc>
                <count>1</count>
                <list>
                    <WASC>
                        <name>
                            <![CDATA[Cross-Site Scripting]]>
                        </name>
                        <url>
<![CDATA[http://projects.webappsec.org/w/page/13246920/WASC]]>
                        </url>
                        <code>8</code>
                    </WASC>
                </list>
            </wasc>
            <resultList>
```

```
                <count>1</count>
                <list>
                    <Result>
                        <authentication>false</authentication>
                        <ajax>false</ajax>
                        <payloads>
                            <count>1</count>
                            <list>
                                <PayloadInstance>
                                    <payload>
                                        <![CDATA[@APPEND@/%22%3e%3cimg
%20src%3dq%20onerror%3dalert(9)%3e]]>
                                    </payload>
                                    <request>
                                        <method>
                                            <![CDATA[GET]]>
                                        </method>
                                        <link>
<![CDATA[http://funkytown.vuln.qa.qualys.com/cassium/traversal/page_48
/%22%3e%3cimg%20src%3dq%20onerror%3dalert(9)%3e]]>
                                        </link>
                                        <headers>
<![CDATA[UmVmZXJlcjogaHR0cDovL2Z1bmt5dG93bi52dWxuLnFhLnF1YWx5cy5jb20vY
2Fzc2l1bS94c3MvDQpDb29raWU6IFBIUFNFU1NJRD00ODlmNTI4ZjUxNWE1MTY3MjM0OTQ
wNzExYTE1MWM0MDsNCg==]]>
                                        </headers>
                                    </request>
                                    <response>
                                        <![CDATA[<html><head><title>We
lcome to page page_48/\"><img src=q
onerror=alert(9)></title></head><body><h1>Welcome to page
page_48/\"><img src=q onerror=alert(9)></h1>Click <a
href='/cassium/traversal/page_49'>here</a> to go to the next
page.Click<a href='/cassium/traversal/page_47'>here</a> to go back to
the previous page.</body></html>]]>
                                    </response>
                    <payloadResponce>
                        <offset>16</offset>
                        <length>62</length>
                    </payloadResponce>
                </PayloadInstance>
            </list>
        </payloads>
    </Result>
</list>
```

```
</resultList>
<severity>5</severity>
<url>
<![CDATA[http://funkytown.vuln.qa.example.com/cassium/traversal/page_4
8/%22%3e%3cimg%20src%3dq%20onerror%3dalert(9)%3e]]>
</url>
<status>ACTIVE</status>
<firstDetectedDate>2017-04-04T06:15:33Z</firstDetectedDate>
<lastDetectedDate>2017-04-04T06:16:20Z</lastDetectedDate>
<lastTestedDate>2017-04-04T06:16:20Z</lastTestedDate>
<timesDetected>3</timesDetected>
<webApp>
    <id>4080112</id>
    <name>
        <![CDATA[web app 1491286489688]]>
    </name>
    <url>
        <![CDATA[http://funkytown.vuln.qa.example.com:80/cassium/xss/]
]>
    </url>
    <tags>
                <count>2</count>
                <list>
                    <Tag>
                        <id>8753812</id>
                        <name>
                            <![CDATA[Multiscan]]>
                        </name>
                    </Tag>
                    <Tag>
                        <id>9029017</id>
                        <name>
                            <![CDATA[TagWebapp1]]>
                        </name>
                    </Tag>
                </list>
    </tags>
</webApp>
<isIgnored>true</isIgnored>
<ignoredReason>FALSE_POSITIVE</ignoredReason>
<ignoredBy>
    <id>6717940</id>
    <username>user_john</username>
    <firstName>
        <![CDATA[John]]>
    </firstName>
    <lastName>
        <![CDATA[Doe]]>
    </lastName>
</ignoredBy>
<ignoredDate>2018-09-06T06:15:44Z</ignoredDate>
<ignoredComment>
    <![CDATA[Test comment]]>
</ignoredComment>
<retest/>
</Finding>
</data>
</ServiceResponse>
```

Sample - Get details of finding

You can fetch details of a finding using uniqueId.

API request

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

XML 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/3.0
/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>132990</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
            <qid>150004</qid>
            <name>
                <![CDATA[Path-Based Vulnerability]]>
            </name>
            <type>VULNERABILITY</type>
            <findingType>QUALYS</findingType>
            <group>PATH</group>
            <cwe>
```

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

Sample - Groups for Information Gathered Issues

Let us view the two groups for issues of type Information Gathered:

- Diagnostic IG (general information about the scan)

- Weakness IG (issues that are security weakness or conflict with best practices)

The response accordingly reflects to which group the issue belongs.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/finding/713223"
```

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/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>713223</id>
            <uniqueId>8c9c933f-04f1-f77e-e053-294f2c0ab892</uniqueId>
            <qid>150014</qid>
            <name>
                <![CDATA[External Form Actions Discovered]]>
            </name>
            <type>INFORMATION_GATHERED</type>
            <findingType>QUALYS</findingType>
            <group>IG_DIAG</group>
            <resultList>
                <count>1</count>
                <list>
                 ....
                </tags>
            </webApp>
        </Finding>
    </data>
</ServiceResponse>
```

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

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

- SSL Data with Certificate Fingerprint

- SSL Data with Prop

- SSL Data with Kex

- SSL Data with Ciphers

The finding you view could include one or multiple issues for an issue type that is listed above. The name tag indicates the type of the issue.

API request

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

XML response (SSL Data with Certificate Fingerprint)

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>581856</id>
            <uniqueId>d6a88c61-fcda-4f46-9767-1d8cb521d953</uniqueId>
            <qid>86002</qid>
            <name>
                <![CDATA[SSL Certificate - Information]]>
            </name>
            <type>INFORMATION_GATHERED</type>
            <findingType>QUALYS</findingType
               ...
                <sslDataInfoList>
                    <list>
                        <SSLDataInfo>
                            <certificateFingerprint>291126AC8ED272F71E
DF06E5B76BBECD1C811769D4FE988DE95FF848AFEBCF6A</certificateFingerprint
>
                        </SSLDataInfo>
                    </list>
                </sslDataInfoList>
            </sslData>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - View Finding Details using Finding ID

This sample shows you the vulnerability fixed date for a finding.

API request

```
curl -n -u "USERNAME:PASSWORD"
"<qulys_base_url>/qps/rest/3.0/get/was/finding/5550534"
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="
    <qualys_base_url>/qps/xsd/3.0/was/finding.
xsd">
        <responseCode>SUCCESS</responseCode>
        <count>1</count>
        <data>
            <Finding>
                <id>5550534</id>
                <uniqueId>0e0f5a34-4226-4a84-bec2-
cc1b0e7f8464</uniqueId>
                <qid>150023</qid>
                <detectionScore>50</detectionScore>
                <name>
                    <![CDATA[Directory Listing]]>
                </name>
                <type>VULNERABILITY</type>
                <potential>false</potential>
                <findingType>QUALYS</findingType>
                <group>PATH</group>
                <cwe>
                    <count>1</count>
                    <list>
                        <long>548</long>
                    </list>
                </cwe>
                <owasp>
                    <count>1</count>
                    <list>
                        <OWASP>
                            <name>
                                <![CDATA[Broken Access Control]]>
                            </name>
                            <url>
                                <![CDATA[https://owasp.org/Top10/A01_2
021-
Broken_Access_Control/]]>
                            </url>
                            <code>1</code>
                        </OWASP>
                    </list>
                </owasp>
                <wasc>
                    <count>1</count>
                    <list>
                        <WASC>
                            <name>
                                <![CDATA[DIRECTORY INDEXING]]>
                            </name>
                            <url>
```

```
                                <![CDATA[http://projects.webappsec.org
/w/page/13246922/WASC]]>
                            </url>
                            <code>16</code>
                        </WASC>
                    </list>
                </wasc>
                <resultList>
                    <count>1</count>
                    <list>
                        <Result>
                            <authentication>false</authentication>
                            <accessPath>
                                <count>3</count>
                                <list>
                                    <Url>
                                        <![CDATA[http://10.11.68.26/]]
>

                                    </Url>
                                    <Url>
                                        <![CDATA[https://10.11.68.26/g
xmail]]>
                                    </Url>
                                    <Url>
                                        <![CDATA[https://10.11.68.26/S
WAGGERAPI]]>
                                    </Url>
                                </list>
                            </accessPath>
                            <ajax>false</ajax>
                            <payloads>
                                <count>1</count>
                                <list>
                                    <PayloadInstance>
                                        <request>
                                            <method>
                                                <![CDATA[GET]]>
                                            </method>
                                            <link>
                                                <firstDetectedDate>202
4-02-08T10:49:48Z</firstDetectedDate>
                                                <lastDetectedDate>2024
-02-08T10:49:48Z</lastDetectedDate>
                                                <lastTestedDate>2024-
02-08T18:37:08Z</lastTestedDate>
```

```
                                                <fixedDate>2024-02-
08T18:37:08Z</fixedDate>
                                                <timesDetected>1</time
sDetected>
                                                <webApp>
                                                    <id>20018939</id>
                                                    <name>
                                                        <![CDATA[Edite
d Name-SRD-Test Progression Ravi New Web App12]]>
                                                    </name>
                                                    <url>
                                                        <![CDATA[http:
//10.11.68.26]]>
                                                    </url>
                                                    <tags>
                                                        <count>4</coun
t>

                                                        <list>
                                                            <Tag>
                                                                <id>26
941212</id>

                                                                <name>
                                                                    <!
[CDATA[SDCheck]]>
                                                                </name
>
                                                            </Tag>
......
```

XSD

<platform API server>/qps/xsd/3.0/was/finding.xsd

Ignore Findings

/qps/rest/3.0/ignore/was/finding
/qps/rest/3.0/ignore/was/finding/<id>

[POST]

Ignore findings for a web application which is in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access" and "Ignore Vulnerabilities" permission. The output includes findings for web applications in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) ID of the finding (WebAppVuln, WebAppIg, or WebAppSensitiveContent).

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

qid	(integer) Qualys ID assigned to the detection.
name	(text) Name of the detection finding.

type	(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.
url	(text) URL of the web application on which the finding was detected.
webApp.tags.id	(date) ID of the tag associated with the web application on which the finding was detected.
webApp.tags.name	(text) Name of the tag associated with the web application on which the finding was detected.
status	(keyword) Status of the finding: NEW, ACTIVE, REOPENED, PROTECTED and FIXED.
patch	(integer-long) Use WAF to protect against vulnerabilities by installing virtual patches.
webApp.id	(integer) ID of the web application on which the finding was detected.
webApp.name	(text) Name of the web application on which the finding was detected.
severity	(integer) Severity of the finding.
externalRef	(string) Tip - Use operator IS EMPTY for findings with empty external references.
ignoredDate	(date) The date on which the finding was marked to ignore.
ignoredReason	(keyword) The reason for which the finding is ignored: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE
group	(keyword) XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM

reactivateDate	(date) Specify the date after which the ignored finding should be re-activated. The date/time is specified in YYYY-MM-DD format.
reactivateIn	(integer) Specify the number of days after which the ignored finding should be reactivated. Note: reactivateDate and reactivateIn are mutually exclusive parameters and cannot be used together. You can use only either of them for a finding.
owasp.name	(text) Name of the OWASP vulnerability.
owasp.code	(integer) Code associated with the OWASP vulnerability
wasc.name	(text) Name of the vulnerability.
wasc.code	(integer) Code of the vulnerability.
cwe.id	(integer) ID associated with CWE.
firstDetectedDate	(date) The date when the finding was first detected in the web application,
lastDetectedDate	(date) The date when the finding was last detected in the web application.
lastTestedDate	(date) The date when the finding was last tested in the web application.
timesDetected	(integer) The count indicating the number of times the finding was detected.
severity level	(integer) The severity associated with the finding:1,2,3,4,5

Sample - Ignore a specific finding

API request

```
curl -n -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/ignore/was/finding/16451956
69"
```

Request POST data

```
<ServiceRequest>
  <data>
          <Finding>
                <id>1645195669</id>
               <ignoredReason>FALSE_POSITIVE</ignoredReason>
                <ignoredComment>test</ignoredComment>
           </Finding>
        </data>
</ServiceRequest>
```

XML 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/3.0
/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Finding>
      <id>1645195669</id>
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
    </Finding>
  </data>
</ServiceResponse>
```

Sample - Reactivate an ignored finding (date)

API request

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

Request POST data

```
<ServiceRequest>
 <data>
   <Finding>
      <id>927823</id>
      <ignoredReason>FALSE_POSITIVE</ignoredReason>
      <ignoredComment>test</ignoredComment>
      <reactivateDate>2018-11-14</reactivateDate>
   </Finding>
  </data>
</ServiceRequest>
```

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/3.
0/
was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>927823</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - Reactivate an ignored finding (day)

API request

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

Request POST data

```
<ServiceRequest>
   <data>
     <Finding>
      <id>927913</id>
      <ignoredReason>FALSE_POSITIVE</ignoredReason>
      <ignoredComment>test</ignoredComment>
      <reactivateIn>1</reactivateIn>
    </Finding>
  </data>
</ServiceRequest>
```

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/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>927913</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - Ignore multiple findings

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="NOT EQUALS">1231056</Criteria>
    <Criteria field="type" operator="NOT
EQUALS">INFORMATION_GATHERED</Criteria>
  </filters>
  <data>
    <Finding>
      <ignoredReason>FALSE_POSITIVE</ignoredReason>
      <ignoredComment>test</ignoredComment>
    </Finding>
  </data>
</ServiceRequest>
```

Note : When you are trying to ignore findings, make sure that type of finding is passed in data is not of INFORMATION_GATHERED type as they cannot be ignored. This can be ensured by using type not equals INFORMATION_GATHERED tag when using NOT EQUALS, GREATER or LESSER operator.

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/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>27</count>
    <data>
        <Finding>
            <id>1231057</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
        </Finding>
        <Finding>
            <id>1231058</id>
            <uniqueId>5a2c4d51-5d28-2b92-e053-2943720a32ab</uniqueId>
        </Finding>
        <Finding>
            <id>1231059</id>
            <uniqueId>4a2c4d51-8d28-2b92-e053-2943720a16ab</uniqueId>
        </Finding>
        <Finding>
            <id>1231060</id>
            <uniqueId>3a2c4d51-9d28-2b92-e053-2943720a90ab</uniqueId>
        </Finding>
       …
</data>
</ServiceResponse>
```

Sample - Ignore finding using uniqueId

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

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <Finding>
            <ignoredReason>FALSE_POSITIVE</ignoredReason>
            <ignoredComment>test</ignoredComment>
        </Finding>
    </data>
</ServiceRequest>
```

XML 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/3.0
/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>132990</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
        </Finding>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/finding.xsd

Activate Findings

/qps/rest/3.0/activate/was/finding

[POST]

Activate ignored findings for a web application which is in the user’s scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "Ignore Vulnerabilities" permission. The output includes findings for web applications in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) ID of the finding (WebAppVuln, WebAppIg, or WebAppSensitiveContent).

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

qid	(integer) Qualys ID assigned to the detection.
name	(text) Name of the detection finding.

type	(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.
url	(text) URL of the web application on which the finding was detected.
webApp.tags.id	(date) ID of the tag associated with the web application on which the finding was detected.
webApp.tags.name	(text) Name of the tag associated with the web application on which the finding was detected.
status	(keyword) Status of the finding: NEW, ACTIVE, REOPENED, PROTECTED and FIXED.
patch	(integer-long) Use WAF to protect against vulnerabilities by installing virtual patches.
webApp.id	(integer) ID of the web application on which the finding was detected.
webApp.name	(text) Name of the web application on which the finding was detected.
severity	(integer) Severity of the finding.
externalRef	(string) Tip - Use operator IS EMPTY for findings with empty external references.
ignoredDate	(date) The date on which the finding was marked to ignore.
ignoredReason	(keyword) The reason for which the finding is ignored: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE
group	(keyword) XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM

owasp.name	(text) Name of the OWASP vulnerability.
owasp.code	(integer) Code associated with the OWASP vulnerability
wasc.name	(text) Name of the vulnerability.
wasc.code	(integer) Code of the vulnerability.
cwe.id	(integer) ID associated with CWE.
firstDetectedDate	(date) The date when the finding was first detected in the web application,
lastDetectedDate	(date) The date when the finding was last detected in the web application.
lastTestedDate	(date) The date when the finding was last tested in the web application.
timesDetected	(integer) The count indicating the number of times the finding was detected.
severity level	(integer) The severity associated with the finding:1,2,3,4,5

Sample - Activate all ignored findings

API request

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

XML 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/3.0
/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>3</count>
  <data>
    <Finding>
      <id>1613225669</id>
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
    </Finding>
    <Finding>
      <id>1613255669</id>
      <uniqueId>9a2c4d41-6d21-2b92-e054-3943720a65ab</uniqueId>
    </Finding>
    <Finding>
      <id>1645195669</id>
      <uniqueId>7a2c4d31-5d28-2b92-e055-4943720a51ab</uniqueId>
    </Finding>
  </data>
</ServiceResponse>
```

Sample - Activate specific finding

API request

```
curl -n -u "USERNAME:PASSWORD"
"qualysapi.qualys.com/qps/rest/3.0/activate/was/finding/1613255669"
```

XML 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/3.0
/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Finding>
      <id>1613255669</id>
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
    </Finding>
  </data>
</ServiceResponse>
```

Sample - Activate a finding using uniqueId

API request

```
curl -n -u "USERNAME:PASSWORD"
"qualysapi.qualys.com/qps/rest/3.0/activate/was/finding/1613255669"
```

XML 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/3.0
/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Finding>
      <id>1613255669</id>
      <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
    </Finding>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/finding.xsd

Edit Finding Severity

/qps/rest/3.0/editSeverity/was/finding

/qps/rest/3.0/editSeverity/was/finding/<id>

[POST]

Edit severity level of the given findings.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "Ignore Vulnerabilities" permission. User must have access to web application which belongs to given WebAppVuln id. The output includes findings for web applications in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) ID of the finding (WebAppVuln, WebAppIg, or WebAppSensitiveContent).

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

new severity level	(integer) {1,2,3,4,5}

comments	(text) User comments.

Sample - Edit severity level

Edit severity for single finding.

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <Finding>
            <id>647</id>
            <severityComment>Test comment API</severityComment>
            <severity>2</severity>
        </Finding>
    </data>
</ServiceRequest>
```

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/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>647</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - Edit severity for multiple findings

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <Finding>
            <severityComment>test comment api</severityComment>
            <severity>2</severity>
        </Finding>
    </data>
    <filters>
           <Criteria field="id" operator="IN">183, 645</Criteria>
       </filters>
</ServiceRequest>
```

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/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <Finding>
            <id>645</id>
            <uniqueId>6a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
        </Finding>
        <Finding>
            <id>183</id>
            <uniqueId>5a2c4d31-5d28-2b92-e055-4943720a51ab</uniqueId>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - Edit severity of a finding using uniqueId

As every uniqueId is unique, using uniqueId, you could edit the severity of a finding.

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <Finding>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
            <severityComment>Test comment API</severityComment>
            <severity>3</severity>
        </Finding>
    </data>
</ServiceRequest>
```

XML 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/3.0
/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>132990</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
        </Finding>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/finding.xsd

Restore Findings Severity

/qps/rest/3.0/restoreSeverity/was/finding

/qps/rest/3.0/restoreSeverity/was/finding/<id>

[POST]

Restore severity level of the given findings.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "Ignore Vulnerabilities" permission. User must have access to web application which belongs to given WebAppVuln id. The output includes findings for web applications in the user's scope.

Input Parameters

The element “id” (integer) is required, where “id” identifies a finding (WebAppVuln, WebAppIg, or WebAppSensitiveContent).

Click here for available operators

Sample - Restore severity level

API request

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

Request POST data

```
 <ServiceRequest>
        <data>
           <Finding>
                <id>6034</id>
           </Finding>
        </data>
   </ServiceRequest>
```

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/3.
0/was/finding.xsd">
        <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>6034</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - Restore for multiple findings

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
           <Criteria field="id" operator="IN">645,183</Criteria>
       </filters>
</ServiceRequest>
```

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/3.
0/was/finding.xsd">
 <responseCode>SUCCESS</responseCode>
    <count>2</count>
    <data>
        <Finding>
            <id>645</id>
            <uniqueId>6a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
        </Finding>
        <Finding>
            <id>183</id>
            <uniqueId>5a2c4d31-5d28-2b92-e055-4943720a51ab</uniqueId>
        </Finding>
    </data>
</ServiceResponse>
```

Sample - Restore severity of a finding using uniqueId

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

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <Finding>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
            <severityComment>Restoring default
severity</severityComment>
            <severity>3</severity>
        </Finding>
    </data>
</ServiceRequest>
```

XML 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/3.0
/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>132990</id>
            <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
        </Finding>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/finding.xsd

Retest Findings

/qps/rest/3.0/retest/was/finding

/qps/rest/3.0/retest/was/finding/<id>

[POST]

You can now easily retest the findings for individual vulnerabilities using Finding API to test the selected finding. Only potential vulnerabilities, confirmed vulnerabilities and sensitive contents are available for retest.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "WAS.VULN.RETEST" permission. The output includes findings for web applications in the user's scope.

Input Parameters

The element “id” (integer) is required, where “id” identifies a finding (WebAppVuln, WebAppIg, or WebAppSensitiveContent).

Click here for available operators

Sample - Retest Finding using XML Request

API request

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

Request POST data

```
 <ServiceRequest>
        <data>
           <Finding>
                <id>1728792</id>
           </Finding>
        </data>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/portal-
api/xsd/3.0/was/finding.xsd" >
  <responseCode>SUCCESS</responseCode>
   <count>1</count>
   <data>
      <Finding>
         <id>1728792</id>
         <uniqueId>2a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
      </Finding>
   </data>
</ServiceResponse>
```

Sample - Using Finding ID

API request

```
curl -n -u "USERNAME:PASSWORD"
"qualysapi.qualys.com/qps/rest/3.0/retest/was/finding/1728792"
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/portal-
api/xsd/3.0/was/finding.xsd"
   <responseCode>SUCCESS</responseCode>
   <count>1</count>
   <data>
      <Finding>
         <id>1728792</id>
         <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
         </Finding>
   </data>
</ServiceResponse>
```

Sample - Retest a finding using uniqueId

API request

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

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/portal-
api/xsd/3.0/was/finding.xsd"
 <responseCode>SUCCESS</responseCode>
  <count>1</count>
   <data>
     <Finding><id>1728792</id></Finding>
     <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>
   </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/finding.xsd

Retrieve Finding Retest Status

/qps/rest/3.0/retestStatus/was/finding/{id}

[POST]

Retrieves the retest status for a finding. You can use the retest status to automate the scanning and retesting processes. The API returns one of these statuses: NO_RETEST, UNDER_RETEST, RETESTED, CANCELING, and CANCELED.

Permissions required - You must have the WAS module enabled. You must have the "API access" and "Access WAS module" permissions. You must have the View permission.

Input Parameters

The API supports POST method. The Input parameters are id or uniqueId. We support optional filters that are available for the Search Finding API.

Parameter	Description
id	(integer) ID of the finding (WebAppVuln or WebAppSensitiveContent).
uniqueId	(value) The 36-bit unique id assigned to the finding.

Sample - Retrieve retest status for a finding

Let us retrieve the retest status of a finding with ID 2730074.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
"https://qualysapi.qualys.com/qps/rest/3.0/retestStatus/was/finding/27
30074"
```

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/3.
0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Finding>
            <id>2774812</id>
            <uniqueId>af45db08-80c6-4527-a48a-9759450b21a2</uniqueId>
            <retest>
                <retestStatus>RETESTED</retestStatus>
                <retestedDate>2020-10-30T09:03:11Z</retestedDate>
                <findingStatus>Finding has been
detected</findingStatus>
                <reason>Finding was confirmed</reason>
            </retest>
        </Finding>
    </data>
</ServiceResponse>
```

WAS Findings in XML Report

Findings in all WAS reports in XML format are Base64 encoded starting with version 3.1. Findings include vulnerability detections, information gathered and sensitive content.

Did you build clients using WAS version 3.0 or earlier? If yes, please update your clients so that WAS findings data is processed accurately.

Tell me about Base64 encoded findings

All findings reported for scan and web applications are base64 encoded in XML. This includes:

- Actual contents of the response

- If evidence in response is highlighted, the evidence contents

- Information gathered data

Base64 encoded data usually will have the attribute set to “base64=true”. For example:

```
<FINDING>
  <PAYLOAD><![CDATA[uid=%00%3Cscript%3E_q%3Drandom(X157105156Y1Z)%3C%2
Fscript
%3E]]></PAYLOAD>
  <RESULT base64="true"><![CDATA[Cl9mZWVkKCgKCgpbCiI=]]></RESULT>
</FINDING>
If the “base64=true attribute” is not set, the value will be in plain
text. For example:
<FINDING>   <PAYLOAD><![CDATA[uid=%00%3Cscript%3E_q%3Drandom(X15710515
6Y1Z)%3C%2Fscript
%3E]]></PAYLOAD>
  <RESULT><![CDATA[_feed(("]]></RESULT>
</FINDING>
```

Which WAS reports show findings?

- WAS v3 Scan Results

- Web Application Report

- Web Application Scan Report

WAS v3 Scan Results

Vulnerability and Sensitive Content findings
WasScan/vulns/list/WasScanVuln/instances/list/WasScanVulnInstance/payloads/list/WasScanVulnPayload/result
WasScan/sensitiveContents/list/WasScanSensitiveContent/instances/list/ WasScanSensitiveContentInstance/payloads/list/WasScanSensitiveContentPayload/result

Sample WAS v3 Scan Results XML

```
<WasScanVuln>
  <qid>150001</qid>
  <title><![CDATA[Reflected Cross-Site Scripting (XSS)
Vulnerabilities]]></title>  <uri><![CDATA[http://myuri.apps.com/613460
625329/feed.gtl?uid=%22'%3E%3Cqss%20a%3DX157
105156Y1Z%3E]]></uri>
  <param>uid</param>
  <instances>
    <count>1</count>
    <list>
      <WasScanVulnInstance>
        <authenticated>false</authenticated>
        <payloads>
          <count>4</count>
          <list>
            <WasScanVulnPayload>            <payload><![CDATA[uid=%00%
3Cscript%3E_q%3Drandom(X157105156Y1Z)%3C%2Fscript%3E]]>
            </payload>
            <result base64="true">
<![CDATA[Cl9mZWVkKCgKCgpbCiI]]></result>
            </WasScanVulnPayload>
            <WasScanVulnPayload>            <payload><![CDATA[uid=%22'
%3E%3Cqss%20a%3DX157105156Y1Z%3E]]></payload>
            <result
base64="true">            <![CDATA[Cl9mZWVkKCgKCgpbCiIiJyZndDsmbHQ7cXN
zIGE9WDE1NzEwNTE1NlkxWiZndDsiCgpdCgoKCikpCg]]></result>
            </WasScanVulnPayload>
            <WasScanVulnPayload>            <payload><![CDATA[uid=%00%
3Cscript%3E_q%3Drandom(X157201836Y1Z)%3C%2Fscript%3E]]>
            </payload>
            <result
base64="true"><![CDATA[Cl9mZWVkKCgKCgpbCiI]]></result>
            </WasScanVulnPayload>
            <WasScanVulnPayload>            <payload><![CDATA[uid=%22'
%3E%3Cqss%20a%3DX157201836Y1Z%3E]]></payload>
            <result
base64="true">            <![CDATA[Cl9mZWVkKCgKCgpbCiIiJyZndDsmbHQ7cXN
zIGE9WDE1NzIwMTgzNlkxWiZndDsiCgpdCgoKCikpCg]]></result>
            </WasScanVulnPayload>
          </list>
        </payloads>
      </WasScanVulnInstance>
    </list>
  </instances>
</WasScanVuln>
```

Information Gathered findings
WasScan/igs/list/WasScanIg/data

Sample WAS v3 Scan Results XML

```
<INFO>
  <QID>150044</QID>
  <TITLE><![CDATA[Login Form Is Not Submitted Via HTTPS]]></TITLE>
  <RESULT base64="true">
<![CDATA[RGVmYXVsdCBmb3JtIGFjdGlvbiBkb2VzIG5vdCBzdWJtaXQgdmlhIFNTTDoga
HR0cDovL2dvb2ds
ZS1ncnV5ZXJlLmFwcHNwb3QuY29tLzYxMzQ2MDYyNTMyOS9sb2dpbgo=]]></RESULT>
</INFO>
```

Vulnerability and Sensitive Content findings
WAS_WEBAPP_REPORT/RESULTS/WEB_APPLICATION/VULNERABILITY _LIST /VULNERABILITY/ PAYLOADS/PAYLOAD/RESPONSE/CONTENTS

WAS_WEBAPP_REPORT/RESULTS/WEB_APPLICATION/SENSITIVE_CONTENT_LIST/ SENSITIVE_CONTENT/PAYLOADS/PAYLOAD/RESPONSE/CONTENTS

WAS_WEBAPP_REPORT/RESULTS/WEB_APPLICATION/VULNERABILITY_LIST/ VULNERABILITY/PAYLOADS/PAYLOAD/RESPONSE/EVIDENCE

WAS_WEBAPP_REPORT/RESULTS/WEB_APPLICATION/SENSITIVE_CONTENT_LIST/ SENSITIVE_CONTENT/PAYLOADS/PAYLOAD/RESPONSE/EVIDENCE

Sample WAS v3 Scan Results XML

```
<VULNERABILITY>
  <ID>5943</ID>
  <QID>150001</QID>
<URL><![CDATA[http://myuri.apps.com/app/xss/0/1/0/xss.php?s='%20onEven
t%3dX146470180Y1Z%20]]></URL>
  <PARAM><![CDATA[s]]></PARAM>
  <AUTHENTICATION>Not Required</AUTHENTICATION>
  <STATUS>NEW</STATUS>
  <FIRST_TIME_DETECTED>2011-12-30T09:57:39Z</FIRST_TIME_DETECTED>
  <LAST_TIME_DETECTED>2011-12-30T09:57:39Z</LAST_TIME_DETECTED>
  <LAST_TIME_TESTED>2011-12-30T09:57:39Z</LAST_TIME_TESTED>
  <TIMES_DETECTED>1</TIMES_DETECTED>
  <PAYLOADS>
    <PAYLOAD>
      <NUM>1</NUM>    <PAYLOAD><![CDATA[s='%20onEvent%3dX146470180Y1Z%
20]]></PAYLOAD>
      <REQUEST/>
      <RESPONSE>
        <CONTENTS
base64="true"><![CDATA[bGQiJmd0OyZsdDsmbHQ7L3NwYW4mZ3Q7ID0mZ3Q7ICZsdDt
zcGFuIGNsYXNzPSJib2xkIiZndDsmYW1wO2x0OyZsdDsvc3BhbiZndDsmbHQ7YnImZ3Q7C
iZsdDsvZGl2Jmd0OwombHQ7L2RpdiZndDsKJmx0O2JyJmd0OwombHQ7ZGl2IGNsYXNzPSJ
wYXlsb2FkcyImZ3Q7Ck91dHB1dCBmcm9tIHJlcXVlc3QgJmx0O3NwYW4gY2xhc3M9ImJvb
GQiJmd0Oy9jYXNzaXVtL3hzcy5waHA/dmFyaWFudD0wJmFtcDtxcz0xJmFtcDtmPTAmYW1
wO3M9JyUyMG9uRXZlbnQlM2RYMTQ2NDcwMTgwWTFaJTIwJmx0Oy9zcGFuJmd0OwombHQ7Y
nImZ3Q7CiZsdDthIGhyZWY9J1wnIG9uRXZlbnQ9WDE0NjQ3MDE4MFkxWiAnJmd0O3NhbXB
sZSBsaW5rJmx0Oy9hJmd0OwombHQ7L2RpdiZndDsKJmx0O3NjcmlwdCZndDttYWluKCkmb
HQ7L3NjcmlwdCZndDsKJmx0Oy9ib2R5Jmd0OwombHQ7L2h0bWwmZ3Q7]]></CONTENTS>
      </RESPONSE>
    </PAYLOAD>
  </PAYLOADS>
    <IGNORED>false</IGNORED>
</VULNERABILITY>
```

Information Gathered findings

WAS_WEBAPP_REPORT/RESULTS/WEB_APPLICATION/
INFORMATION_GATHERED_LIST/
INFORMATION_GATHERED/DATA

```
<INFORMATION_GATHERED_LIST>
  <INFORMATION_GATHERED>
    <ID>1529</ID>
    <QID>6</QID>
    <FIRST_TIME_DETECTED>2011-12-30T09:57:39Z</FIRST_TIME_DETECTED>
    <LAST_TIME_DETECTED>2011-12-30T09:57:39Z</LAST_TIME_DETECTED>
    <LAST_TIME_TESTED>2011-12-30T09:57:39Z</LAST_TIME_TESTED>
<DATA
base64="true"><![CDATA[I3RhYmxlCklQX2FkZHJlc3MgSG9zdF9uYW1lCgoxMC4xMC4
yNi43NyBmdW5reXR vd24udnVsbi5x
YS5xdWFseXMuY29tCg==]]></DATA>
  </INFORMATION_GATHERED>
  <INFORMATION_GATHERED>
    <ID>1532</ID>
    <QID>150031</QID>
    <FIRST_TIME_DETECTED>2011-12-30T09:57:39Z</FIRST_TIME_DETECTED>
    <LAST_TIME_DETECTED>2011-12-30T09:57:39Z</LAST_TIME_DETECTED>
    <LAST_TIME_TESTED>2011-12-30T09:57:39Z</LAST_TIME_TESTED>
<DATA
base64="true"><![CDATA[VGltZW91dCByZWFjaGVkIGluIElQQyBjb25uZWN0aW9uIHR
vIFdlYktpdC4gSmF
2YVNjcmlwdCBz
dXBwb3J0IGRpc2FibGVkIGluOmVQaGFzZUNyYXdsCkNyYXdsIGNvbXBsZXRlZCB3aXRoIF
dlYktp dC4K]]></DATA>
</INFORMATION_GATHERED>
```

Vulnerability and Sensitive Content findings

WAS_SCAN_REPORT/RESULTS/VULNERABILITY_LIST/VULNERABILITY/ PAYLOADS/ PAYLOAD/RESPONSE/CONTENTS
WAS_SCAN_REPORT/RESULTS/SENSITIVE_CONTENT_LIST/ SENSITIVE_CONTENT/ PAYLOADS/PAYLOAD/RESPONSE/CONTENTS
WAS_SCAN_REPORT/RESULTS/VULNERABILITY_LIST/VULNERABILITY/ PAYLOADS/ PAYLOAD/RESPONSE/EVIDENCE
WAS_SCAN_REPORT/RESULTS/SENSITIVE_CONTENT_LIST/ SENSITIVE_CONTENT/ PAYLOADS/PAYLOAD/RESPONSE/EVIDENCE
Information Gathered findings
WAS_SCAN_REPORT/RESULTS/INFORMATION_GATHERED_LIST/ INFORMATION_GATHERED/DATA

Reference: Findings

The <OptionProfile> element includes sub elements used to define an option profile. A reference of these elements is provided below. An asterisk * indicates a complex element.

Parameter	Description
id	(integer) ID of the finding (WebAppVuln, WebAppIg, or WebAppSensitiveContent).

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

qid	(integer) Qualys ID assigned to the detection.
name	(text) Name of the detection finding.
type	(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.
url	(text) URL of the web application on which the finding was detected.
webApp.tags.id	(integer) ID of the tag associated with the web application on which the finding was detected.
webApp.tags.name	(text) Name of the tag associated with the web application on which the finding was detected.
status	(keyword) Status of the finding: NEW, ACTIVE, REOPENED, PROTECTED and FIXED.

patch	(integer-long) Use WAF to protect against vulnerabilities by installing virtual patches.
webApp.id	(integer) ID of the web application on which the finding was detected.
webApp.name	(text) Name of the web application on which the finding was detected.
severity	(integer) Severity of the finding.
externalRef	(string) Tip - Use operator IS EMPTY for findings with empty external references.
ignoredDate	(date) The date on which the finding was marked to ignore.
ignoredReason	(keyword) The reason for which the finding is ignored: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE
group	(keyword) XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM
owasp.name	(text) Name of the OWASP vulnerability.
owasp.code	(integer) Code associated with the OWASP vulnerability
wasc.name	(text) Name of the vulnerability.
wasc.code	(integer) Code of the vulnerability.
cwe.id	(integer) ID associated with CWE.
firstDetectedDate	(date) The date when the finding was first detected in the web application.
lastDetectedDate	(date) The date when the finding was last detected in the web application.

lastTestedDate	(date) The date when the finding was last tested in the web application.
timesDetected	(integer) The count indicating the number of times the finding was detected.
severity level	(integer) The severity associated with the finding:1,2,3,4,5

Configuration

Option Profiles

Option Profile Count

/qps/rest/3.0/count/was/optionprofile

[GET] [POST]

Returns the total number of option profiles in the user's scope. Input elements are optional and are used to filter the number of option profiles included in the count.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The count includes web applications in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) The ID of the option profile.
name	(text) The name given to the option profile.
tags	Filter by tags applied.
tags.id	(integer) ID of the tag assigned to option profile.
tags.name	(text) Tag name assigned to option profile.

createdDate	(date) The date when the option profile was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the option profile was updated in WAS, in UTC date/time format.
usedByWebApps	(boolean) Web applications used/not used by the option profile.
usedBySchedules	(boolean) Scan schedules used/not used by the option profile.
owner.id	(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the option profile.
owner.name	(text) Full name of the user who created the option profile.
owner.username	(text) Username of the owner who created the option profile. (like user_ab3).

Sample - Count - no criteria (GET)

API request

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

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>30</count>
</ServiceResponse>
```

Sample - Count - criteria (POST)

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
      <Criteria field="id"
operator="IN">832265669,832295669,832285669</Criteria>
      <Criteria field="name" operator="CONTAINS">OP</Criteria>
      <Criteria field="tags" operator="NONE"></Criteria>
      <Criteria field="createdDate" operator="LESSER">2017-09-
09</Criteria>
      <Criteria field="updatedDate" operator="LESSER">2017-09-
09</Criteria>
  </filters>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/optionprofile.xsd

Search Option Profiles

/qps/rest/3.0/search/was/optionprofile

[POST]

Returns a list of option profiles which are in the user’s scope. Action logs are not included in the output.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”. The Output includes option profiles in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) The ID of the option profile.
name	(text) The name given to the option profile.
tags	Filter by tags applied.
tags.id	(integer) ID of the tag assigned to option profile.
tags.name	(text) Tag name assigned to option profile.
createdDate	(date) The date when the option profile was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the option profile was updated in WAS, in UTC date/time format.
usedByWebApps	(boolean) Web applications used/not used by the option profile.

usedBySchedules	(boolean) Scan schedules used/not used by the option profile.
owner.id	(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the option profile.
owner.name	(text) Full name of the user who created the option profile.
owner.username	(text) Username of the owner who created the option profile. (like user_ab3).

Sample - Search - criteria (POST)

API request

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

Request POST data

```
<ServiceRequest>
  <filters>
    <Criteria field="id"
operator="IN">832265669,832295669,832285669</Criteria>
        <Criteria field="name" operator="CONTAINS">OP</Criteria>
        <Criteria field="tags" operator="NONE"></Criteria>
        <Criteria field="createdDate" operator="LESSER">2017-09-
09</Criteria>
        <Criteria field="updatedDate" operator="LESSER">2017-09-
09</Criteria>
  </filters>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <OptionProfile>
      <id>832285669</id>
      <name><![CDATA[My Option Profile]]></name>
      <owner>
        <id>8792415669</id>
        <username>user_ww</username>
        <firstName><![CDATA[Walter]]></firstName>
        <lastName><![CDATA[White]]></lastName>
      </owner>
      <tags>
        <count>0</count>
      </tags>
      <createdDate>2017-09-08T23:16:07Z</createdDate>
      <updatedDate>2017-09-08T23:16:07Z</updatedDate>
    </OptionProfile>
  </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/optionprofile.xsd

Get Option Profile Details

/qps/rest/3.0/get/was/optionprofile/<id>

[GET]

View details for an option profile which is in the user's scope. See "Search option profiles" to find a record ID to use as input.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The Output includes option profiles in the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies an option profile.

Click here for available operators

Samples

Sample - Get details of an option profile (GET)

Sample - Get details on option profile with SmartScan enabled (GET)

Sample - View details to know if action URI is enabled

Sample - Get details of an Option Profile with customized scan intensity (GET)

Sample - Get details of an option profile with enhanced crawling enabled (GET)

Sample - Get details of an option profile to know the detection scope (GET)

Sample - Get details of an option profile (GET)

API request

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/optionprofile/83226
5669"
```

XML 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/3.0
/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>832265669</id>
      <name><![CDATA[My Option Profile]]></name>
      <owner>
        <id>8792415669</id>
        <username>user_walter</username>
        <firstName><![CDATA[Walter]]></firstName>
        <lastName><![CDATA[White]]></lastName>
      </owner>
      <isDefault>false</isDefault>
      <tags>
        <count>0</count>
      </tags>
      <formSubmission>BOTH</formSubmission>
      <maxCrawlRequests>300</maxCrawlRequests>
      <timeoutErrorThreshold>200</timeoutErrorThreshold>
      <unexpectedErrorThreshold>20</unexpectedErrorThreshold>
      <parameterSet>
        <id>0</id>
        <name><![CDATA[Initial Parameters]]></name>
      </parameterSet>
      <ignoreBinaryFiles>false</ignoreBinaryFiles>
      <performance>LOW</performance>
      <bruteforceOption>MINIMAL</bruteforceOption>
      <comments>
        <count>2</count>
        <list>
          <Comment>
            <contents><![CDATA[some comments]]></contents>
            <author>
              <id>200639085669</id>
              <username>user_walter</username>
            </author>
          </Comment>
          <Comment>
            <contents><![CDATA[some more comments]]></contents>
            <author>
              <id>200639085669</id>
              <username>user_walter</username>
            </author>
          </Comment>
        </list>
      </comments>
      <sensitiveContent>
        <creditCardNumber>false</creditCardNumber>
        <socialSecurityNumber>false</socialSecurityNumber>
      </sensitiveContent>
      <createdDate>2017-09-08T22:03:01Z</createdDate>
      <createdBy>
        <id>8792415669</id>
        <username>user_walter</username>
        <firstName><![CDATA[Walter]]></firstName>
        <lastName><![CDATA[White]]></lastName>
      </createdBy>
      <updatedDate>2017-09-08T23:18:28Z</updatedDate>
      <updatedBy>
        <id>8792415669</id>
        <username>user_walter</username>
        <firstName><![CDATA[Walter]]></firstName>
        <lastName><![CDATA[White]]></lastName>
      </updatedBy>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Sample - Get details on option profile with SmartScan enabled (GET)

Want to use SmartScan? This feature must be enabled for your subscription. We can help you with this quickly - just contact your Technical Account Manager or Qualys Support.

API request

```
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/optionprofile/46733
3"
```

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>467333</id>
            <name>
                <![CDATA[My Option Profile]]>
            </name>
            <owner>
                <id>4354</id>
                <username>user_aril</username>
                <firstName>
                    <![CDATA[Ari]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>300</maxCrawlRequests>
            <timeoutErrorThreshold>100</timeoutErrorThreshold>
            <unexpectedErrorThreshold>300</unexpectedErrorThreshold>
            <parameterSet>
                <id>15601</id>
                <name>
                    <![CDATA[Test Paramset]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>false</ignoreBinaryFiles>
            <smartScanSupport>true</smartScanSupport>
            <smartScanDepth>10</smartScanDepth>
            <performance>LOW</performance>
            <bruteforceOption>MINIMAL</bruteforceOption>
            <comments>
                <count>0</count>
            </comments>
            <sensitiveContent>
                <creditCardNumber>false</creditCardNumber>
                <socialSecurityNumber>false</socialSecurityNumber>
            </sensitiveContent>
            <createdDate>2017-03-23T21:15:47Z</createdDate>
            <createdBy>
                <id>4354</id>
                <username>user_aril</username>
                <firstName>
                    <![CDATA[Ari]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </createdBy>
            <updatedDate>2017-03-23T21:15:47Z</updatedDate>
            <updatedBy>
                <id>4354</id>
                <username>user_aril</username>
                <firstName>
                    <![CDATA[Ari]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </updatedBy>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - View details to know if action URI is enabled

Example: View the option profile details for the web application with ID #171683 to check if action URI is enabled or disabled.

API request

```
curl -u "USERNAME:PASSWORD" " -X GET -H "Content-type: text/xml"
"https://qualysapi.qualys.com/portal-
api/rest/3.0/get/was/optionprofile/176683"
```

XML response

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

```
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/portal-
api/xsd/3.0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>176683</id>
            <name>
                <![CDATA[My Option Profile - with action URI]]>
            </name>
            <owner>
                <id>336390</id>
                <username>john_doe</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Doe]]>
                </lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>200</maxCrawlRequests>
            <timeoutErrorThreshold>22</timeoutErrorThreshold>
            <unexpectedErrorThreshold>50</unexpectedErrorThreshold>
            <userAgent>
                <![CDATA[Mozilla/5.0 (Windows NT 6.2;
WOW64)AppleWebKit
                   /537.36 (KHTML, like Gecko) Chrome/27.0.1453.116
                   Safari/537.36]]>
            </userAgent>
            <parameterSet>
                <id>0</id>
                <name>
                    <![CDATA[Initial Parameters]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>true</ignoreBinaryFiles>
         <includeActionUriInFormId>true</includeActionUriInFormId>
            <smartScanSupport>false</smartScanSupport>
            <performance>LOW</performance>
            <bruteforceOption>DISABLED</bruteforceOption>
            <comments>
                <count>1</count>
                <list>
                    <Comment>
                        <contents>
                            <![CDATA[User Comment]]>
                        </contents>
                        <createdDate>2017-11-
18T15:59:55Z</createdDate>
                    </Comment>
                </list>
            </comments>
           ...
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Get details of an Option Profile with customized scan intensity (GET)

Let us get details of an Option Profile with customized scan intensity.

API request

```
curl -u "USERNAME:PASSWORD" " -X GET -H "Content-type: text/xml"
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/optionprofile/16085
60"
```

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>1608560</id>
            <name>
                <![CDATA[Update Option Profile with Custom Scan
Intensity]]>
            </name>
            …
            <smartScanSupport>false</smartScanSupport>
            <customPerformance>
                <numOfHttpThreads>10</numOfHttpThreads>
                <delayBetweenRequests>20</delayBetweenRequests>
            </customPerformance>
            <bruteforceOption>MINIMAL</bruteforceOption>
            …
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Get details of an option profile with enhanced crawling enabled (GET)

API request

```
curl -u "USERNAME:PASSWORD" " -X GET -H "Content-type: text/xml"
 "https://qualysapi.qualys.com/qps/rest/3.0/get/was/optionprofile/7768
3"
```

XML 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/3.0
/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>77683</id>
            <name>
                <![CDATA[Sample Option Profile]]>
            </name>
            <owner>
                <id>337590</id>
                <username>user_john</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Doe]]>
                </lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>300</maxCrawlRequests>
            <timeoutErrorThreshold>100</timeoutErrorThreshold>
            <unexpectedErrorThreshold>300</unexpectedErrorThreshold>
            <parameterSet>
                <id>0</id>
                <name>
                    <![CDATA[Initial Parameters]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>false</ignoreBinaryFiles>
            <includeActionUriInFormId>false</includeActionUriInFormId>
            <enhancedCrawling>true</enhancedCrawling>
            <smartScanSupport>false</smartScanSupport>
            ...
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Get details of an option profile to know the detection scope (GET)

API request

```
curl -u "USERNAME:PASSWORD" " -X GET -H "Content-type: text/xml"
 "https://qualysapi.qualys.com/qps/rest/3.0/get/was/optionprofile/7768
3"
```

XML 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/3.0
/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>77683</id>
            <name>
                <![CDATA[Sample Option Profile]]>
            </name>
            <owner>
```

```
                <id>337590</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>1000</maxCrawlRequests>
            <timeoutErrorThreshold>100</timeoutErrorThreshold>
            <unexpectedErrorThreshold>300</unexpectedErrorThreshold>
            <parameterSet>
                <id>0</id>
                <name>
                    <![CDATA[Initial Parameters]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>true</ignoreBinaryFiles>
            <includeActionUriInFormId>false</includeActionUriInFormId>
            <enhancedCrawling>false</enhancedCrawling>
            <smartScanSupport>true</smartScanSupport>
            <smartScanDepth>5</smartScanDepth>
            <performance>LOW</performance>
            <bruteforceOption>MINIMAL</bruteforceOption>
            <detection>
                <detectionScope>EVERYTHING</detectionScope>
            </detection>
            <comments>
                <count>0</count>
            </comments>
            ...
            </updatedBy>
        </OptionProfile>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/optionprofile.xsd

Create a new Option Profile

/qps/rest/3.0/create/was/optionprofile

[POST]

Create a new option profile.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and "Create Option Profile".

Input Parameters

The element “name” (text) and "OptionProfile" is required, where “name” is option profile name.

Click here for available operators

Samples

Create - minimum criteria (POST)

Create - multiple criteria (POST)

Create - disable error threshold values, set to 0 (POST)

Create - enable SmartScan (POST)

Create - enable action URI (POST)

Create - associate pre-defined detection category (POST)

Create an option profile with XSS Power Mode detection scope (POST)

Create - Enabling XSS Payloads for standard scan

Create - custom scan intensity (POST)

Create - Enhanced Crawling enabled (POST)

Create - Everything as detection scope

Create - SSL/TLS and Certificate issues

Sample - Create - minimum criteria (POST)

Create a new option profile with the name “My Option Profile - with defaults”. The default option profile settings are assigned automatically.

API request

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

Request POST data

```
<ServiceRequest>
   <data>
      <OptionProfile>
         <name><![CDATA[My Option Profile - with defaults]]></name>
      </OptionProfile>
   </data>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>832265669</id>
      <name><![CDATA[My Option Profile - with defaults]]></name>
      <owner>
        <id>8792415669</id>
        <username>user_alex</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <isDefault>false</isDefault>
      <tags>
```

```
        <count>0</count>
      </tags>
      <formSubmission>BOTH</formSubmission>
      <maxCrawlRequests>300</maxCrawlRequests>
      <timeoutErrorThreshold>20</timeoutErrorThreshold>
      <unexpectedErrorThreshold>48</unexpectedErrorThreshold>
      <parameterSet>
        <id>0</id>
        <name><![CDATA[Initial Parameters]]></name>
      </parameterSet>
      <ignoreBinaryFiles>false</ignoreBinaryFiles>
      <performance>LOW</performance>
      <bruteforceOption>MINIMAL</bruteforceOption>
      <comments>
        <count>0</count>
      </comments>
      <sensitiveContent>
        <creditCardNumber>false</creditCardNumber>
        <socialSecurityNumber>false</socialSecurityNumber>
      </sensitiveContent>
      <createdDate>2018-09-08T22:03:01Z</createdDate>
      <createdBy>
        <id>8792415669</id>
        <username>user_alex</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </createdBy>
      <updatedDate>2018-09-08T22:03:01Z</updatedDate>
      <updatedBy>
        <id>8792415669</id>
        <username>user_alex</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </updatedBy>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Sample - Create - multiple criteria (POST)

Create a new option profile with the name “My Option Profile - All Fields”. The "name" setting is required in the request data, other settings are optional.

API request

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

Request POST data

```
<ServiceRequest>
   <data>
      <OptionProfile>
         <name><![CDATA[My Option Profile - All Fields]]></name>
         <timeoutErrorThreshold>22</timeoutErrorThreshold>
         <unexpectedErrorThreshold>50</unexpectedErrorThreshold>
         <formSubmission>BOTH</formSubmission>
         <maxCrawlRequests>200</maxCrawlRequests>
         <performance>LOW</performance>
         <bruteforceOption>USER_DEFINED</bruteforceOption>
         <parameterSet><id>15669</id></parameterSet>
         <isDefault>true</isDefault>
         <ignoreBinaryFiles>true</ignoreBinaryFiles>
         <userAgent><![CDATA[Mozilla/5.0 (Windows NT 6.2; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116
Safari/537.36]]></userAgent>
         <tags><set><Tag><id>75521225669</id></Tag></set></tags>
         <sensitiveContent>
         <customContents>zip code</customContents>
         </sensitiveContent>
         <comments>
            <set>
                 <Comment>

<contents><![CDATA[Some Comment]]></contents>
               </Comment>
            </set>
         </comments>
         <bruteforceList>
            <id>74005669</id>
         </bruteforceList>
         <detection>
            <includedSearchLists>
               <set>
                   <SearchList>
                        <id>3496185669</id>
                   </SearchList>
               </set>
            </includedSearchLists>
            <excludedSearchLists>
               <set>
                  <SearchList>
                     <id>3496175669</id>
                  </SearchList>
                  <SearchList>
                     <id>3496165669</id>
                  </SearchList>
               </set>
            </excludedSearchLists>
         </detection>
      </OptionProfile>
   </data>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>832275669</id>
      <name><![CDATA[My Option Profile - All Fields]]></name>
      <owner>
        <id>8792415669</id>
        <username>user_cindy</username>
        <firstName><![CDATA[Cindy]]></firstName>
        <lastName><![CDATA[Green]]></lastName>
      </owner>
      <isDefault>true</isDefault>
      <tags>
        <count>1</count>
        <list>
          <Tag>
            <id>75521225669</id>
            <name><![CDATA[Business Units]]></name>
          </Tag>
        </list>
      </tags>
      <formSubmission>BOTH</formSubmission>
      <maxCrawlRequests>200</maxCrawlRequests>
      <timeoutErrorThreshold>22</timeoutErrorThreshold>
      <unexpectedErrorThreshold>50</unexpectedErrorThreshold>
      <userAgent><![CDATA[Mozilla/5.0 (Windows NT 6.2; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116
Safari/537.36]]></userAgent>
      <parameterSet>
        <id>15669</id>
        <name><![CDATA[Custom Parameters]]></name>
      </parameterSet>
      <ignoreBinaryFiles>true</ignoreBinaryFiles>
      <performance>LOW</performance>
      <bruteforceOption>USER_DEFINED</bruteforceOption>
      <bruteforceList>
        <id>74005669</id>
        <name><![CDATA[BFL]]></name>
      </bruteforceList>
      <detection>
        <includedSearchLists>
          <count>1</count>
          <list>
            <SearchList>
              <id>3496185669</id>
            </SearchList>
          </list>
        </includedSearchLists>
        <excludedSearchLists>
          <count>2</count>
          <list>
            <SearchList>
              <id>3496175669</id>
            </SearchList>
            <SearchList>
              <id>3496165669</id>
            </SearchList>
          </list>
        </excludedSearchLists>
      </detection>
      <comments>
        <count>1</count>
        <list>
          <Comment>
            <contents><![CDATA[Some Comment]]></contents>
          </Comment>
        </list>
      </comments>
      <sensitiveContent>
        <creditCardNumber>false</creditCardNumber>
        <socialSecurityNumber>false</socialSecurityNumber>
        <customContents>zip code</customContents>
      </sensitiveContent>
      <createdDate>2017-09-08T22:31:06Z</createdDate>
      <createdBy>
        <id>8792415669</id>
        <username>user_cindy</username>
        <firstName><![CDATA[Cindy]]></firstName>
        <lastName><![CDATA[Green]]></lastName>
      </createdBy>
      <updatedDate>2017-09-08T22:31:07Z</updatedDate>
      <updatedBy>
        <id>8792415669</id>
        <username>user_cindy</username>
        <firstName><![CDATA[Cindy]]></firstName>
        <lastName><![CDATA[Green]]></lastName>
      </updatedBy>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Sample - Create - disable error threshold values, set to 0 (POST)

API request

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

Request POST data

```
<ServiceRequest>
   <data>
      <OptionProfile>
         <name><![CDATA[My OP - with no threshold
specified]]></name>
         <timeoutErrorThreshold>0</timeoutErrorThreshold>
         <unexpectedErrorThreshold>0</unexpectedErrorThreshold>
      </OptionProfile>
   </data>
</ServiceRequest>
```

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>453133</id>
            <name>
                <![CDATA[My OP - with no threshold specified]]>
            </name>
            <owner>
                <id>4354</id>
                <username>user_amy</username>
                <firstName>
                    <![CDATA[Amy]]>
                </firstName>
                <lastName>
                    <![CDATA[Kim]]>
                </lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>300</maxCrawlRequests>
            <parameterSet>
                <id>0</id>
                <name>
                    <![CDATA[Initial Parameters]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>false</ignoreBinaryFiles>
            <performance>LOW</performance>
            <bruteforceOption>MINIMAL</bruteforceOption>
            <comments>
                <count>0</count>
            </comments>
            <sensitiveContent>
                <creditCardNumber>false</creditCardNumber>
                <socialSecurityNumber>false</socialSecurityNumber>
            </sensitiveContent>
            <createdDate>2017-11-07T01:29:24Z</createdDate>
            <createdBy>
...
```

Sample - Create - enable SmartScan (POST)

Want to use SmartScan? This feature must be enabled for your subscription. We can help you with this quickly - just contact your Technical Account Manager or Qualys Support.

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <OptionProfile>
            <name>My Option Profile</name>
            <smartScanSupport>true</smartScanSupport>
            <smartScanDepth>10</smartScanDepth>
        </OptionProfile>
    </data>
</ServiceRequest>
```

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>467333</id>
            <name>
                <![CDATA[My Option Profile]]>
            </name>
            <owner>
                <id>4354</id>
                <username>user_aril</username>
                <firstName>
                    <![CDATA[Ari]]>
                </firstName>
                <lastName>
                    <![CDATA[Smith]]>
                </lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>300</maxCrawlRequests>
            <timeoutErrorThreshold>100</timeoutErrorThreshold>

            <unexpectedErrorThreshold>300</unexpectedErrorThreshold>
            <parameterSet>
                <id>15601</id>
                <name>
                    <![CDATA[Test Paramset]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>false</ignoreBinaryFiles>
            <smartScanSupport>true</smartScanSupport>
            <smartScanDepth>10</smartScanDepth>
            <performance>LOW</performance>
            <bruteforceOption>MINIMAL</bruteforceOption>>
...
```

Sample - Create - enable action URI (POST)

Create a new option profile with the name “My Option Profile” to include action URI. The default option profile settings are assigned automatically.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml"-X "POST"--
data-binary@-
```

```
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/optionprofile"
< file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
   <data>
      <OptionProfile>
          <name>
             <![CDATA[My Option Profile]]>
          </name>
          <timeoutErrorThreshold>22</timeoutErrorThreshold>
          <unexpectedErrorThreshold>50</unexpectedErrorThreshold>
          <formSubmission>BOTH</formSubmission>
          <maxCrawlRequests>200</maxCrawlRequests>
          <performance>LOW</performance>
          <bruteforceOption>DISABLED</bruteforceOption>
          <isDefault>true</isDefault>
          <ignoreBinaryFiles>true</ignoreBinaryFiles>
         <includeActionUriInFormId>true</includeActionUriInFormId>
          <userAgent>
          <![CDATA[Mozilla/5.0 (Windows NT 6.2;
WOW64)AppleWebKit/537.36
            (KHTML, like Gecko) Chrome/27.0.1453.116 Safari/537.36]]>
          </userAgent>
          <sensitiveContent>
          <customContents>zip code</customContents>
          </sensitiveContent>
          <comments>
          <set>
          <Comment>
          <contents>
          <![CDATA[This is a test comment.]]>
          </contents>
          </Comment>
          </set>
          </comments>
        </OptionProfile>
   </data>
</ServiceRequest>
```

XML response

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

```
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/portal-
api/xsd/3.0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>171683</id>
            <name>
                <![CDATA[My Option Profile - with action URI]]>
            </name>
            <owner>
                <id>336390</id>
                <username>john_doe</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Doe]]>
                </lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>200</maxCrawlRequests>
            <timeoutErrorThreshold>22</timeoutErrorThreshold>
            <unexpectedErrorThreshold>50</unexpectedErrorThreshold>
            <userAgent>
                <![CDATA[Mozilla/5.0 (Windows NT 6.2;
WOW64)AppleWebKit
                   /537.36 (KHTML, like Gecko) Chrome/27.0.1453.116
                   Safari/537.36
                  ]]>
            </userAgent>
            <parameterSet>
                <id>0</id>
                <name>
                    <![CDATA[Initial Parameters]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>true</ignoreBinaryFiles>
         <includeActionUriInFormId>true</includeActionUriInFormId>
            <smartScanSupport>false</smartScanSupport>
            <performance>LOW</performance>
            <bruteforceOption>DISABLED</bruteforceOption>
            <comments>
                <count>1</count>
                <list>
                    <Comment>
                        <contents>
                            <![CDATA[User Comment]]>
                        </contents>
                        <createdDate>2017-11-
18T15:59:55Z</createdDate>
                    </Comment>
                </list>
            </comments>
            <sensitiveContent>
                <creditCardNumber>false</creditCardNumber>
                <socialSecurityNumber>false</socialSecurityNumber>
                <customContents>zip code</customContents>
            </sensitiveContent>
            <createdDate>2017-11-18T15:59:49Z</createdDate>
            <createdBy>
                <id>336390</id>
                <username>john_doe</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Doe]]>
                </lastName>
            </createdBy>
            <updatedDate>2017-11-18T15:59:49Z</updatedDate>
            <updatedBy>
                <id>336390</id>
                <username>john_doe</username>
                <firstName>
                    <![CDATA[John]]>
                </firstName>
                <lastName>
                    <![CDATA[Doe]]>
                </lastName>
            </updatedBy>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Create - associate pre-defined detection category

Create a new option profile and associate pre-defined detection categories with Option Profile.

Element	Description
detectionCategory={Keyword}	We now support the following new detection categories in your option profile: --XSS, in request header --Denial of Service --XSS --Path-Related vulnerabilities --OWASP Top 10 (2017) --Authentication & Session Management --Cross-Site Request Forgery --XML External Entity (XXE) vulnerabilities --Flash-Related vulnerabilities --Information Disclosure --SQL Injection --Clickjacking --SQL Injection, in request header --CMS identification (type, version, and plugins) --Apache vulnerabilities (Struts & other)

	--Uncategorized --CMS vulnerabilities --Open Redirect
	Note: <detectionCategories> is mutually exclusive with <includedSearchLists> and <excludedSearchLists>.

API request

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

Request POST data

```
<ServiceRequest>
   <data>
      <OptionProfile>
      <name>sample option profile with detection category</name>
         <detection>
            <detectionCategories>
               <set>
                  <DetectionCategory>
                     <name>Denial of Service</name>
                  </DetectionCategory>
               </set>
            </detectionCategories>
         </detection>
      </OptionProfile>
   </data>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
```

```
<responseCode>SUCCESS</responseCode>
   <count>1</count>
     <data>
       <count>1</count>
    <data>
        <OptionProfile>
            <id>305785</id>
            <name>
                <![CDATA[Sample option profile with detection
category]]>
            </name>
            <owner>
                <id>2501086</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            ...
            <detection>
                <detectionCategories>
                    <count>1</count>
                    <set>
                        <DetectionCategory>
                            <id>154</id>
                            <name>Denial of Service</name>
                        </DetectionCategory>
                    </set>
                </detectionCategories>
            </detection>
            <comments>
                <count>0</count>
            </comments>
            ...
        </OptionProfile>
    </data>
</ServiceResponse>
```

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

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

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

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

API request

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

Request POST data

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

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>1045129</id>
            <name>
                <![CDATA[Launch XSS Power Mode Scan]]>
            </name>
            <owner>
                <id>412791</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>300</maxCrawlRequests>
            <timeoutErrorThreshold>100</timeoutErrorThreshold>
            <unexpectedErrorThreshold>300</unexpectedErrorThreshold>
            <parameterSet>
                <id>0</id>
                <name>
                    <![CDATA[Initial Parameters]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>false</ignoreBinaryFiles>
            <includeActionUriInFormId>false</includeActionUriInFormId>
            <smartScanSupport>false</smartScanSupport>
            <performance>LOW</performance>
            <bruteforceOption>MINIMAL</bruteforceOption>
            <detection>
                <xssPowerMode>true</xssPowerMode>
            </detection>
            <comments>
                <count>0</count>
            </comments>
            <sensitiveContent>
                <creditCardNumber>false</creditCardNumber>
                <socialSecurityNumber>false</socialSecurityNumber>
            </sensitiveContent>
            <createdDate>2018-07-25T03:45:12Z</createdDate>
            <createdBy>
            <owner>
                <id>412791</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            </createdBy>
            <updatedDate>2018-07-25T03:45:12Z</updatedDate>
```

```
            <updatedBy>
             <owner>
                <id>412791</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            </updatedBy>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Enabling XSS Payloads for standard scan

You can enable comprehensive tests for cross-site scripting vulnerabilities to be executed during our standard scan using the new parameter in option profile. The comprehensive tests includes XSS with exhaustive set of payloads including set of standard payloads. Running a scan with XSS payloads option enabled in the detection scope of standard scan will provide the best assurance that your web application is free from XSS vulnerabilities. However, enabling this option leads to significant increase in the scan time.

Element	Description

enableXssPayloads	(boolean) A flag to indicate if XSS payloads should be enabled or disabled during the scan. If the flag is set to true, comprehensive tests for cross-site scripting vulnerabilities are executed during the scan.
	Example:
	`<detection>`
	`  <detectionScope>CORE</detectionScope>`
	`  <enableXssPayloads>true</enableXssPayloads>`
	`</detection>`

Let us create an option profile to launch a standard scan with comprehensive tests for cross-site scripting vulnerabilities enabled.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
databinary@-
```

```
"https://qualysapi.qualys.com/qps/rest/3.0/create/was/optionprofile" <
file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <data>
      <OptionProfile>
         <name>Sample Option Profile With XSS Payloads</name>
         <detection>
            <detectionScope>CORE</detectionScope>
            <enableXssPayloads>true</enableXssPayloads>
         </detection>
      </OptionProfile>
   </data>
</ServiceRequest>
```

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>81333</id>
            <name>
                <![CDATA[Launch Scan with XSS Payloads enabled]]>
            </name>
            <owner>
                <id>412791</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>300</maxCrawlRequests>
            <timeoutErrorThreshold>100</timeoutErrorThreshold>
            <unexpectedErrorThreshold>300</unexpectedErrorThreshold>
            <parameterSet>
                <id>0</id>
                <name>
                    <![CDATA[Initial Parameters]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>false</ignoreBinaryFiles>
            <includeActionUriInFormId>false</includeActionUriInFormId>
            <enhancedCrawling>false</enhancedCrawling>
            <smartScanSupport>false</smartScanSupport>
            <performance>LOW</performance>
            <bruteforceOption>MINIMAL</bruteforceOption>
            <detection>
                <detectionScope>CORE</detectionScope>
                <enableXssPayloads>true</enableXssPayloads>
            </detection>
            <comments>
                <count>0</count>
            </comments>
            <sensitiveContent>
                <creditCardNumber>false</creditCardNumber>
                <socialSecurityNumber>false</socialSecurityNumber>
            </sensitiveContent>
            <createdDate>2019-10-04T11:11:59Z</createdDate>
            <createdBy>
            <owner>
                <id>412791</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            </createdBy>
            <updatedDate>2018-07-25T03:45:12Z</updatedDate>
            <updatedBy>
             <owner>
                <id>412791</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            </updatedBy>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Create an option profile with custom scan intensity

You can define your custom scan intensity in the option profile and thus control the scan performance accordingly to your configured settings. Using our new parameter <customperformance> you can further configure the number of threads to be used to scan each host and the delay between requests.

API request

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

Request POST data

```
<ServiceRequest>
<data>
    <OptionProfile>
        <name><![CDATA[Option Profile with Custom Scan
Intensity]]></name>
            <customPerformance>
                <numOfHttpThreads>5</numOfHttpThreads>
                <delayBetweenRequests>100</delayBetweenRequests>
            </customPerformance>
        </OptionProfile>
</data>
</ServiceRequest>
```

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/3.
0/was/optionprofile.xsd">
     <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>1608560</id>
            <name>
                <![CDATA[Option Profile with Custom Scan Intensity]]>
            </name>
            …
            <smartScanSupport>false</smartScanSupport>
            <customPerformance>
                <numOfHttpThreads>5</numOfHttpThreads>
                <delayBetweenRequests>100</delayBetweenRequests>
            </customPerformance>
            <bruteforceOption>MINIMAL</bruteforceOption>
…
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Create an option profile with Enhanced Crawling enabled

API request

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

Request POST data

```
<ServiceRequest>
 <data>
  <OptionProfile>
   <name><![CDATA[Sample Option Profile]]></name>
   <enhancedCrawling>true</enhancedCrawling>
 </data>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>78110</id>
            <name><![CDATA[Sample Option Profile]]></name>
            <owner>
```

```
                <id>337590</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>300</maxCrawlRequests>
            <timeoutErrorThreshold>100</timeoutErrorThreshold>
            <unexpectedErrorThreshold>300</unexpectedErrorThreshold>
            <parameterSet>
                <id>0</id>
                <name>
                    <![CDATA[Initial Parameters]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>false</ignoreBinaryFiles>
            <includeActionUriInFormId>false</includeActionUriInFormId>
            <enhancedCrawling>true</enhancedCrawling>
            <smartScanSupport>false</smartScanSupport>
            <performance>LOW</performance>
            <bruteforceOption>MINIMAL</bruteforceOption>
            <detection/>
            <comments>
                <count>0</count>
            </comments>
            ...
                </lastName>
            </updatedBy>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Create - Everything as detection scope

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <OptionProfile>
            <name><![CDATA[Sample Option Profile]]></name>
            <detection>
                <detectionScope>EVERYTHING</detectionScope>
            </detection>
        </OptionProfile>
    </data>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>78744</id>
            <name>
                <![CDATA[Sample Option Profile]]>
            </name>
            <owner>
                <id>337590</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            <formSubmission>BOTH</formSubmission>
            <maxCrawlRequests>300</maxCrawlRequests>
            <timeoutErrorThreshold>100</timeoutErrorThreshold>
            <unexpectedErrorThreshold>300</unexpectedErrorThreshold>
            <parameterSet>
                <id>0</id>
                <name>
                    <![CDATA[Initial Parameters]]>
                </name>
            </parameterSet>
            <ignoreBinaryFiles>false</ignoreBinaryFiles>
            <includeActionUriInFormId>false</includeActionUriInFormId>
            <enhancedCrawling>false</enhancedCrawling>
            <smartScanSupport>false</smartScanSupport>
            <performance>LOW</performance>
            <bruteforceOption>MINIMAL</bruteforceOption>
            <detection>
                <detectionScope>EVERYTHING</detectionScope>
            </detection>
            <comments>
                <count>0</count>
            </comments>
            ...
            </updatedBy>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Create - SSL/TLS and Certificate issues

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

API request

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

Request POST data

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

XML 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/3.0
/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>897483</id>
            <name>
                <![CDATA[My Option Profile - SSL data]]>
            </name>
            <owner>
                <id>412791</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
           <isDefault>false</isDefault>
            <tags>
                <count>0</count>
            </tags>
            ...
            <detection>
                <detectionCategories>
                    <count>1</count>
                    <list>
                        <DetectionCategory>
                            <id>152</id>
                            <name>SSL/TLS and Certificate
issues</name>
                        </DetectionCategory>
                    </list>
                </detectionCategories>
                <enableXssPayloads>false</enableXssPayloads>
            </detection>
            ...
        </OptionProfile>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/optionprofile.xsd

Update an Option Profile

/qps/rest/3.0/update/was/optionprofile/<id>

[POST]

Update an option profile which is in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access" and "Update Option Profile".

Input Parameters

The element "id" (integer) is required, where "id" identifies an option profile. Additional elements are optional and must be supplied in POST XML data. At least one of the following elements must be set: name, isDefault, owner, tags, formSubmission, maxCrawlRequests, userAgent, parameterSet, ignoreBinaryFiles, performance, bruteforceOption, bruteforceList, numberOfAttempts, detection, sensitiveContent, comments.

Click here for available operators

Parameter	Description
id	(integer) The ID of the option profile.
name	(text) The name given to the option profile.
tags	Filter by tags applied.
tags.id	(integer) ID of the tag assigned to option profile.
tags.name	(text) Tag name assigned to option profile.
owner.id	(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the option profile.

owner.name	(text) Full name of the user who created the option profile.
owner.username	(text) Username of the owner who created the option profile. (like user_ab3).
isDefault	Default option profile for the subscription
formSubmission	(keyword) Type of form: None, Post, Get, POST& GET
maxCrawlRequests	Total number of links and forms to follow and test within the scan scope. If performing a Discovery Scan, this is the maximum links that will be crawled, as there will not be any testing performed
userAgent	Stores the browser and OS details.
parameterSet	A parameter set tells us the request parameter settings you would like us to inject into your web applications during scanning. We provide a default one and it is easy to configure more. Once defined just select the parameter set name in your scan's option profile.
ignoreBinaryFiles	If you choose these option files with extension zip, pdf, doc are not scanned.
performance	(keyword) Scan Intensity: LOWEST, LOW, MEDIUM, HIGH, MAXIMUM.
customPerformance*	Configure the custom intensity level for web application scans. Example: <customPerformance> <numOfHttpThreads>10</numOfHttpThreads> <delayBetweenRequests>5</delayBetweenRequ

	ests> </customPerformance> Note: performance and customPerformance are mutually exclusive parameters and cannot be used together. You can use only either of them for an option profile.
numOfHttpThreads	(integer) Number of threads to be used to scan each host. The valid range is from 1 to 10.
delayBetweenRequests	(integer) The duration of delay introduced by WAS in between the scanning engine requests sent to the applications server. The valid range is from 0 to 2000 milliseconds.
bruteforceOption	The level of brute forcing you prefer with options ranging from "Minimal" to "Exhaustive".
bruteforceList	(keyword: User List/SYSTEM LIST) System list: we'll attempt to guess the password for each detected login ID. User list: to select a bruteforce list defined in your account .
numberOfAttempts	The threshold to be reached before stopping the scan. If you deactivate this settings, the scan will keep running no matter how many errors it will find.
detection	(keyword) Select if scans launched with this profile shall perform a full assessment for all WAS detections the engine is able to discover, or if the scan shall focus on the detection of specific vulnerabilities and/or information: Core, Categories, Custom Search list, XSS Power Mode, Everything. If <detectionScope> is present then the detection scope = CORE or EVERYTHING

	Core: Core scope includes vulnerabilities that Qualys considers most common in today's web applications. It does not include all the vulnerabilities that WAS can detect.
	Everything: Everything scope includes all the vulnerabilities that WAS can detect.
	Example:
	
	`<detection>`
	`<detectionScope>EVERYTHING</detectionScope>`
	`</detection>`
	
	If <includedSearchLists> or <excludedSearchLists> are present then the detection scope = CUSTOM
	
	If <detectionCategories> is present then the detection scope = CATEGORY
	
	if <xssPowerMode> is true then the detection scope = XSS
	
	Note: The <includedSearchLists>, <excludedSearchLists>, <detectionCategories>, <xssPowerMode>, <detectionScope> elements are mutually exclusive elements.

enableXssPayloads	(boolean) A flag to indicate if XSS payloads should be enabled or disabled during the scan. If the flag is set to true, comprehensive tests for cross-site scripting vulnerabilities are executed during the scan.
	
	Example:
	
	`<detection>`
	`  <detectionScope>CORE</detectionScope>`
	`  <enableXssPayloads>true</enableXssPayloads>`
	`</detection>`

sensitiveContent	Credit Card Numbers, Social Security Numbers (US), Custom Contents.

keywordsUrlSearch	(text) Specify keywords in the form of strings and regular expressions to search for URL links that contains the specified keyword. Currently, we search for keywords only in the internal links that are found in the crawling phase for target web applications in a Discovery/Vulnerability scan. You can enter a maximum of 10 keywords where each keyword appears on a separate line. A keyword should be 5 to 200 characters long. During a Discovery/Vulnerability scan, we search for these keywords and report all the unique links that contain the specified keywords in the Get Finding Details API output under information gathered QID 150141. Note that we show the crawled links under QID 150009.
enhancedCrawling	(boolean) Improve scan coverage for your web application with the enhanced crawling enabled. We will re-crawl individual directories present in the links which are found during crawling. For example, if the following link is found during crawling: https://www.example.com/foo/abc/xyz/register.php If the enhanced crawling is enabled, it will first make a request to https://www.example.com/foo/abc/xyz and will then remove the directory "xyz/" from the URL and crawl, https://www.example.com/foo/abc/ and later it will further remove "abc/" and will crawl https://www.example.com/foo/. All the links found during this process of removal and re-crawling will get added to the crawl queue thus improving the scan coverage.

comments	User-defined comments.

Samples

Update - minimum criteria (POST)

Update - multiple settings (POST)

Update - owner (POST)

Update - custom threshold values (POST)

Update - disable action URI (POST)

Update - Detection Category (POST)

Update Option Profile for Custom Scan Intensity (POST)

Update an Option Profile to disable enhanced crawling (POST)

Update option profile to change detection scope to Everything

Update Option Profile to enable XSS payload

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

Sample - Update - minimum criteria (POST)

Change the option profile name to “Update Option Profile - title” for option profile ID 832265669.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"--
data-binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/83
2265669"  < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
   <data>
      <OptionProfile>
```

```
         <name><![CDATA[Update Option Profile - title ]]></name>
      </OptionProfile>
   </data>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>832265669</id>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Sample - Update - multiple settings (POST)

Update multiple option profile settings for option profile ID 832275669.

API request

```
url -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"--
data-binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/83
2275669"  < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
   <data>
      <OptionProfile>
         <name><![CDATA[My Option Profile - All Fields]]></name>
         <formSubmission>BOTH</formSubmission>
         <maxCrawlRequests>100</maxCrawlRequests>
         <performance>HIGH</performance>
         <bruteforceOption>USER_DEFINED</bruteforceOption>
         <parameterSet><id>15669</id></parameterSet>
         <isDefault>false</isDefault>
         <ignoreBinaryFiles>false</ignoreBinaryFiles>
         <userAgent><![CDATA[Mozilla/5.0 (Windows NT 6.2; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116
Safari/537.36]]></userAgent>
         <tags><set><Tag><id>75521225669</id></Tag></set></tags>
         <sensitiveContent>
            <customContents>zip code</customContents>
         </sensitiveContent>
         <comments>
            <set>
               <Comment>
                  <contents><![CDATA[Comment 2]]></contents>
               </Comment>
            </set>
         </comments>
         <bruteforceList>
            <id>74005669</id>
         </bruteforceList>
         <detection>
            <includedSearchLists>
               <set>
                  <SearchList>
                     <id>3496185669</id>
                  </SearchList>
               </set>
            </includedSearchLists>
            <excludedSearchLists>
               <set>
                  <SearchList>
                     <id>3496175669</id>
                  </SearchList>
                  <SearchList>
                     <id>3496165669</id>
                  </SearchList>
               </set>
            </excludedSearchLists>
         </detection>
      </OptionProfile>
   </data>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>832275669</id>
    </OptionProfile>
  </data>
  </ServiceRequest>
```

Sample - Update - owner (POST)

Update the option profile owner.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"--
data-binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/12
3456"  < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
   <data>
      <OptionProfile>
         <owner><id>123456</id></owner>
      </OptionProfile>
   </data>
</ServiceRequest>
```

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>123456</id>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Update - custom threshold values (POST)

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"--
data-binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/45
2933"  < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
   <data>
      <OptionProfile>
         <name><![CDATA[My OP - with custom threshold
values]]></name>
         <timeoutErrorThreshold>200</timeoutErrorThreshold>
         <unexpectedErrorThreshold>20</unexpectedErrorThreshold>
      </OptionProfile>
   </data>
</ServiceRequest>
```

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>452933</id>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Update - disable action URI (POST)

Update the Option Profile to disable Action URI.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml"-X "POST"--
data-binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/17
6683"  < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
    <data>
        <OptionProfile>
            <name>
            <![CDATA[My Option Profile - with action URI]]>
            </name>
            <includeActionUriInFormId>false</includeActionUriInFormId>
        </OptionProfile>
    </data>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"xsi:noNamespaceSchemaLocation="http://qualysapi
.qualys.com/portal-api/xsd/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>176683</id>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Sample - Update - Detection Category (POST)

Update the detection scope in the Option Profile.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml"-X "POST"--
data-binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/17
6683"  < file.xml
```

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

Request POST data

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
   <data>
      <OptionProfile>
         <detection>
            <detectionCategories>
               <remove>
                  <DetectionCategory>
                     <name>Denial of Service</name>
                  </DetectionCategory>
               </remove>
               <add>
                 <DetectionCategory>
                     <name>SQL Injection</name>
                  </DetectionCategory>
               </add>
            </detectionCategories>
         </detection>
      </OptionProfile>
   </data>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd
/3.0/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
   <count>1</count>
    <data>
        <OptionProfile>
            <id>305786</id>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Update Option Profile for Custom Scan Intensity (POST)

Let us update an Option Profile with customized scan intensity.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml"-X "POST"--
data-binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/16
08560"  < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
<data>
    <OptionProfile>
        <name><![CDATA[Update Option Profile with Custom Scan
Intensity]]></name>
            <customPerformance>
                <numOfHttpThreads>10</numOfHttpThreads>
                <delayBetweenRequests>20</delayBetweenRequests>
            </customPerformance>
        </OptionProfile>
</data>
</ServiceRequest>
```

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>1608560</id>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Update an Option Profile to disable enhanced crawling (POST)

Let us update an Option Profile with customized scan intensity.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"--
data-binary@-
```

```
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/83
2265669" < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
 <data>
  <OptionProfile>
    <enhancedCrawling>false</enhancedCrawling>
  </OptionProfile>
 </data>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>832265669</id>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Update option profile to change detection scope to Everything

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"--
data-binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/83
2265669" < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
 <data>
   <OptionProfile>
     <detection>
```

```
       <detectionScope>EVERYTHING</detectionScope>
     </detection>
   </OptionProfile>
  </data>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>832265669</id>
        </OptionProfile>
    </data>
</ServiceResponse>
```

Sample - Update Option Profile to enable XSS payload

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"--
data-binary@-
"https://qualysapi.qualys.com/qps/rest/3.0/update/was/optionprofile/16
003" < file.xml
Note: “file.xml” contains the request POST data.
```

Request POST data

```
<ServiceRequest>
  <data>
      <OptionProfile>
         <name>Sample Option Profile With XSS Payloads</name>
         <detection>
            <detectionScope>CORE</detectionScope>
            <enableXssPayloads>true</enableXssPayloads>
         </detection>
      </OptionProfile>
   </data>
</ServiceRequest>
```

XML 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/3.0
/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>16003</id>
        </OptionProfile>
    </data>
</ServiceResponse>
```

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

API request

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

Request POST data

```
<ServiceRequest>
 <data>
  <OptionProfile>
   <detection>
        <detectionCategories>
            <set>
                <DetectionCategory>
                    <name>SSL/TLS and Certificate issues</name>
                </DetectionCategory>
            </set>
        </detectionCategories>
    </detection>
  </OptionProfile>
 </data>
</ServiceRequest>
```

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/3.
0/was/optionprofile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <OptionProfile>
            <id>897483</id>
        </OptionProfile>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/optionprofile.xsd

Delete an Option Profile

/qps/rest/3.0/delete/was/optionprofile/<id>

/qps/rest/3.0/delete/was/optionprofile

[POST]

Delete an option profile that is in the user's scope. Upon success, the output is a list of IDs for the option profiles that were deleted.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access" and "Delete Option Profile".

Input Parameters

Optional elements are used to retrieve option profiles to delete. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Click here for available operators

Parameter	Description
name	(text) The name given to the option profile.
owner	(text) Username of the owner who created the option profile. (like user_ab3).
tags	(text) Filter by tags applied to option profile.
createdDate	(date) The date when the option profile was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the option profile was updated in WAS, in UTC date/time format.
usedByWebApps	(boolean) Web applications used/not used by the option profile.

usedBySchedules	(boolean) Scan schedules used/not used by the option profile.

Sample - Delete specific option profile (POST)

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --
data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/optionprofile/83
4275669"
```

XML 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/3.0
/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <OptionProfile>
      <id>834275669</id>
    </OptionProfile>
  </data>
</ServiceResponse>
```

Sample - Delete multiple option profiles (POST)

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
        <Criteria field="name" operator="CONTAINS">OP</Criteria>
        <Criteria field="updatedDate" operator="LESSER">2017-09-
09</Criteria>
    </filters>
</ServiceRequest>
```

XML response

```
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/3.0
/was/optionprofile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>10</count>
  <data>
    <OptionProfile>
      <id>712265669</id>
    </OptionProfile>
    <OptionProfile>
      <id>752265669</id>
    </OptionProfile>
    <OptionProfile>
      <id>752275669</id>
    </OptionProfile>
    <OptionProfile>
      <id>754265669</id>
    </OptionProfile>
    <OptionProfile>
      <id>812685669</id>
    </OptionProfile>
    <OptionProfile>
      <id>824295669</id>
    </OptionProfile>
    <OptionProfile>
      <id>824305669</id>
    </OptionProfile>
    <OptionProfile>
      <id>830265669</id>
    </OptionProfile>
    <OptionProfile>
      <id>830275669</id>
    </OptionProfile>
    <OptionProfile>
      <id>830285669</id>
    </OptionProfile>
      </data>
  </ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/optionprofile.xsd

Reference: Option Profile

The <OptionProfile> element includes sub elements used to define an option profile. A reference of these elements is provided below. An asterisk * indicates a complex element.

Parameter	Description
id	(integer) The ID of the option profile.
name	(text) The name given to the option profile.
tags	Filter by tags applied.
tags.id	(integer) ID of the tag assigned to option profile.
tags.name	(text) Tag name assigned to option profile.
createdDate	(date) The date when the option profile was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the option profile was updated in WAS, in UTC date/time format.
usedByWebApps	(boolean) Web applications used/not used by the option profile.
usedBySchedules	(boolean) Scan schedules used/not used by the option profile.
owner.id	(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the option profile.
owner.name	(text) Full name of the user who created the option profile.
owner.username	(text) Username of the owner who created the option profile. (like user_ab3).

isDefault	Default option profile for the subscription
formSubmission	(keyword) Type of form: None, Post, Get, POST& GET
maxCrawlRequests	Total number of links and forms to follow and test within the scan scope. If performing a Discovery Scan, this is the maximum links that will be crawled, as there will not be any testing performed
userAgent	Stores the browser and OS details.
parameterSet	A parameter set tells us the request parameter settings you would like us to inject into your web applications during scanning. We provide a default one and it is easy to configure more. Once defined just select the parameter set name in your scan's option profile.
ignoreBinaryFiles	If you choose these option files with extension zip, pdf, doc are not scanned.
performance	(keyword) Scan Intensity: LOWEST, LOW, MEDIUM, HIGH, MAXIMUM.
customPerformance*	Configure the custom intensity level for web application scans. Example: <customPerformance> <numOfHttpThreads>10</numOfHttpThreads> <delayBetweenRequests>5</delayBetweenRequests> </customPerformance> Note: performance and customPerformance are mutually exclusive parameters and cannot be used together. You can use only either of them for an option profile.

numOfHttpThreads	(integer) Number of threads to be used to scan each host. The valid range is from 1 to 10.
delayBetweenRequests	(integer) The duration of delay introduced by WAS in between the scanning engine requests sent to the applications server. The valid range is from 0 to 2000 milliseconds.
bruteforceOption	The level of brute forcing you prefer with options ranging from "Minimal" to "Exhaustive".
bruteforceList	(keyword: User List/SYSTEM LIST) System list: we'll attempt to guess the password for each detected login ID. User list: to select a bruteforce list defined in your account .
numberOfAttempts	The threshold to be reached before stopping the scan. If you deactivate this settings, the scan will keep running no matter how many errors it will find.
detection	(keyword) Select if scans launched with this profile shall perform a full assessment for all WAS detections the engine is able to discover, or if the scan shall focus on the detection of specific vulnerabilities and/or information: Core, Categories, Custom Search list, XSS Power Mode, Everything. If <detectionScope> is present then the detection scope = CORE or EVERYTHING Core: Core scope includes vulnerabilities that Qualys considers most common in today's web applications. It does not include all the vulnerabilities that WAS can detect. Everything: Everything scope includes all the vulnerabilities that WAS can detect. Example:

	`<detection>`
	`<detectionScope>EVERYTHING</detectionScope>`
	`</detection>`
	
	If <includedSearchLists> or <excludedSearchLists> are present then the detection scope = CUSTOM
	
	If <detectionCategories> is present then the detection scope = CATEGORY
	
	if <xssPowerMode> is true then the detection scope = XSS
	
	Note: The <includedSearchLists>, <excludedSearchLists>, <detectionCategories>, <xssPowerMode>, <detectionScope> elements are mutually exclusive elements.

sensitiveContent	Credit Card Numbers, Social Security Numbers (US), Custom Contents.
keywordsUrlSearch	(text) Specify keywords in the form of strings and regular expressions to search for URL links that contains the specified keyword. Currently, we search for keywords only in the internal links that are found in the crawling phase for target web applications in a Discovery/Vulnerability scan. You can enter a maximum of 10 keywords where each keyword appears on a separate line. A keyword should be 5 to 200 characters long. During a Discovery/Vulnerability scan, we search for these keywords in the internal links and report all the unique links that contain the specified keywords in the Get Finding Details API output under information gathered QID 150141. Note that we show the crawled links under QID 150009.
enhancedCrawling	(boolean) Improve scan coverage for your web application with the enhanced crawling enabled.

	We will re-crawl individual directories present in the links which are found during crawling. For example, if the following link is found during crawling: https://www.example.com/foo/abc/xyz/register.php If the enhanced crawling is enabled, it will first make a request to https://www.example.com/foo/abc/xyz and will then remove the directory "xyz/" from the URL and crawl, https://www.example.com/foo/abc/ and later it will further remove "abc/" and will crawl https://www.example.com/foo/. All the links found during this process of removal and re-crawling will get added to the crawl queue thus improving the scan coverage.
comments	User-defined comments.

DNS Override

DNS Override Count

/qps/rest/3.0/count/was/dnsoverride/

[GET] [POST]

Returns the total number of DNS overrides in the user's scope. Input elements are optional and are used to filter the number of option profiles included in the count.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The count includes web applications in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) The ID of the DNS override.
name	(text) The name given to the DNS override.
tags	Filter by tags applied.
tags.id	(integer) ID of the tag assigned to DNS override.
tags.name	(text) Tag name assigned to DNS override.
createdDate	(date) The date when the DNS override was created in WAS, in UTC date/time format.

updatedDate	(date) The date when the DNS override was updated in WAS, in UTC date/time format.
owner.id	(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the DNS override.
owner.name	(text) Full name of the user who created the DNS override.
owner.username	(text) Username of the owner who created the DNS override. (like user_ab3).

Sample - Count (POST)

API request

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

Request POST data

```
<ServiceRequest>
     <filters>
           <Criteria field="name" operator="CONTAINS">Test
API</Criteria>
     </filters>
</ServiceRequest>
```

XML 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/3.0
/was/dnsoverride.xsd">
<responseCode>SUCCESS</responseCode>
    <count>6</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/dnsoverride.xsd

Search DNS Override

/qps/rest/3.0/search/was/dnsoverride/

[POST]

Returns a list of DNS overrides which are in the user’s scope. Action logs are not included in the output.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”. The Output includes DNS overrides in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) The ID of the DNS override.
name	(text) The name given to the DNS override.
tags	Filter by tags applied.
tags.id	(integer) ID of the tag assigned to DNS override.
tags.name	(text) Tag name assigned to DNS override.
createdDate	(date) The date when the DNS override was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the DNS override was updated in WAS, in UTC date/time format.

owner.id	(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the DNS override.
owner.name	(text) Full name of the user who created the DNS override.
owner.username	(text) Username of the owner who created the DNS override. (like user_ab3).

Sample - Search - criteria (POST)

API request

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

Request POST data

```
<ServiceRequest>
     <filters>
           <Criteria field="name" operator="CONTAINS">Test
API</Criteria>
     </filters>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/xsd/3.0/was
/dnsoverride.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>6</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <DnsOverride>
            <id>56420</id>
            <name>
                <![CDATA[Test API DNS Record]]>
            </name>
            <owner>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <tags>
                <count>0</count>
            </tags>
            <createdDate>2019-08-12T13:33:04Z</createdDate>
            <updatedDate>2019-08-12T13:33:04Z</updatedDate>
        </DnsOverride>
        <DnsOverride>
            <id>56422</id>
            <name>
                <![CDATA[Test API Dns Record1]]>
            </name>
            <owner>
                <id>1056860</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <tags>
                <count>0</count>
            </tags>
            <createdDate>2019-08-12T13:58:59Z</createdDate>
            <updatedDate>2019-08-12T13:58:59Z</updatedDate>
        </DnsOverride>
        <DnsOverride>
            <id>56423</id>
            <name>
                <![CDATA[Test API Dns Record2]]>
            </name>
            <owner>
                <id>1056860</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <tags>
                <count>2</count>
            </tags>
            <createdDate>2019-08-12T15:30:24Z</createdDate>
            <updatedDate>2019-08-12T15:30:30Z</updatedDate>
        </DnsOverride>
        <DnsOverride>
            <id>56621</id>
            <name>
                <![CDATA[Test API Dns Record3]]>
            </name>
            <owner>
                <id>1056860</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <tags>
                <count>2</count>
            </tags>
            <createdDate>2019-08-12T23:03:53Z</createdDate>
            <updatedDate>2019-08-12T23:03:59Z</updatedDate>
        </DnsOverride>
        <DnsOverride>
            <id>56820</id>
            <name>
                <![CDATA[Test API Dns Record3-Updated]]>
            </name>
            <owner>
                <id>1056860</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <tags>
                <count>0</count>
            </tags>
            <createdDate>2019-08-13T00:07:37Z</createdDate>
            <updatedDate>2019-08-16T14:10:18Z</updatedDate>
        </DnsOverride>
        <DnsOverride>
            <id>57020</id>
            <name>
                <![CDATA[Test API Dns Record4]]>
            </name>
            <owner>
                <id>1056860</id>
                <username>user_john</username>
                <firstName><![CDATA[John]]></firstName>
                <lastName><![CDATA[Doe]]></lastName>
            </owner>
            <tags>
                <count>1</count>
            </tags>
            <createdDate>2019-08-19T16:25:05Z</createdDate>
            <updatedDate>2019-08-22T12:35:40Z</updatedDate>
        </DnsOverride>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/dnsoverride.xsd

Get DNS Override Details

/qps/rest/3.0/get/was/dnsoverride/<id>

[GET]

View details for an DNS override which is in the user's scope. See "Search DNS overrides" to find a record ID to use as input.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access". The Output includes DNS overrides in the user's scope.

Input Parameters

The element "id" (integer) is required, where "id" identifies an option profile.

Click here for available operators

Sample - Get details of an option profile (GET)

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

API request

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

XML 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/3.0
/was/dnsoverride.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <DnsOverride>
            <id>57020</id>
            <name>
                <![CDATA[Test API DNS Record4]]>
            </name>
            ...
            <mappings>
                <count>3</count>
                <list>
                    <DnsMapping>
                        <hostName>host_1</hostName>
                        <ipAddress>1.2.3.7</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_2</hostName>
                        <ipAddress>1.2.3.5</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_3</hostName>
                        <ipAddress>1.2.3.5</ipAddress>
                    </DnsMapping>
                </list>
            </mappings>
        </DnsOverride>
    </data>
</ServiceResponse>
```

Create DNS Override

/qps/rest/3.0/create/was/dnsoverride

[POST]

Create a new DNS Override.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”.

Input Parameters

The element “name” (text) and "mappings" is required, where “name” is name of the DNS override.

Click here for available operators

Parameter	Description
name	(text) The name given to the DNS override.

DnsMapping (keyword)	Use to configure the DNS override setting through API. You need to specify the hostname or FQDN and the corresponding IP address to be preferred for scanning.
	Example:
	`<set>`
	`   <DnsMapping>`
	`      <hostName>test</hostName>`
	`       <ipAddress>2.3.4.5</ipAddress>`
	`   </DnsMapping>`
	`</set>`
	When you create a new DNS override, ensure:
	-Name (Required): Name should be unique.

-Tags: The tag id should be valid and in scope of current user. Use only <Set> tag.

-Mappings (Required): Each mapping must have hostName and IpAddress in valid format. Use only <Set> tag.

-Comments: Only <Set> with 1 comment is allowed with maximum length 2048 characters.

Sample - Create DNS Override (POST)

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <DnsOverride>
            <name><![CDATA[DNS Record]]></name>
            <mappings>
                <set>
                    <DnsMapping>
                        <hostName>host_1</hostName>
                        <ipAddress>2.3.4.5</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_2</hostName>
                        <ipAddress>1.2.3.4</ipAddress>
                    </DnsMapping>
                </set>
            </mappings>
            <tags>
                <set>
                    <Tag>
                        <id>8993614</id>
                    </Tag>
                    <Tag>
```

```
                        <id>8876615</id>
                    </Tag>
                </set>
            </tags>
         </DnsOverride>
    </data>
</ServiceRequest>
```

XML 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/3.0
/was/dnsoverride.xsd">
     <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <DnsOverride>
            <id>57220</id>
            <name>
                <![CDATA[DNS Record]]>
            </name>
            ...
            <mappings>
                <count>2</count>
                <list>
                    <DnsMapping>
                        <hostName>host_1</hostName>
                        <ipAddress>2.3.4.5</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_2</hostName>
                        <ipAddress>1.2.3.4</ipAddress>
                    </DnsMapping>
                </list>
            </mappings>
        </DnsOverride>
    </data>
</ServiceResponse>
```

Update an DNS Override

/qps/rest/3.0/update/was/dnsoverride

[POST]

Update an DNS override which is in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access".

Input Parameters

Click here for available operators

Parameter	Description
name	(text) The name given to the DNS override.

DnsMapping (keyword)	Use to configure the DNS override setting through API. You need to specify the hostname or FQDN and the corresponding IP address to be preferred for scanning.
	Example:
	`<set>`
	`   <DnsMapping>`
	`      <hostName>test</hostName>`
	`       <ipAddress>2.3.4.5</ipAddress>`
	`   </DnsMapping>`
	`</set>`
	When you update an DNS override, ensure:
	-Name: In case of name update, the updated name should be unique.
	-Id is required.

-At lease one of the following should be present other than id: Name, owner, tags, comments, mappings

-Tags: The <set> and <Add>/ <Removed> tags are mutually exclusive. Either use <set> or <Add> and <Removed>.

- Mappings: The <set> and <Add>/ <Removed> tags are mutually exclusive. Either use <set> or <Add> and <Removed>.

Sample - Update DNS Override (POST)

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <DnsOverride>
            <name><![CDATA[DNS Record]]></name>
            <mappings>
                <set>
                    <DnsMapping>
                        <hostName>host_1</hostName>
                        <ipAddress>2.3.4.5</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_2</hostName>
                        <ipAddress>1.2.3.4</ipAddress>
                    </DnsMapping>
                </set>
            </mappings>
            <tags>
                <set>
                    <Tag>
                        <id>8993614</id>
                    </Tag>
                    <Tag>
                        <id>8876615</id>
                    </Tag>
                </set>
            </tags>
         </DnsOverride>
    </data>
</ServiceRequest>
```

XML 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/3.0
/was/dnsoverride.xsd">
   <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <DnsOverride>
            <id>57020</id>
        </DnsOverride>
    </data>
</ServiceResponse>
```

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

API request

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

Request POST data

```
<ServiceRequest>
    <data>
        <DnsOverride>
            <name><![CDATA[DNS Record]]></name>
            <mappings>
                <remove>
                    <DnsMapping>
                        <hostName>host_1</hostName>
                        <ipAddress>1.2.3.4</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_2</hostName>
                        <ipAddress>1.2.3.6</ipAddress>
                    </DnsMapping>
                </remove>
                <add>
                    <DnsMapping>
                        <hostName>host_3</hostName>
                        <ipAddress>1.2.3.5</ipAddress>
                    </DnsMapping>
                    <DnsMapping>
                        <hostName>host_4</hostName>
                        <ipAddress>1.2.3.7</ipAddress>
                    </DnsMapping>
                </add>
            </mappings>
            <tags>
                <set>
                    <Tag>
                        <id>8993614</id>
                    </Tag>
                    <Tag>
                        <id>8876615</id>
                    </Tag>
                </set>
            </tags>
         </DnsOverride>
    </data>
</ServiceRequest>
```

XML 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/3.0
/was/dnsoverride.xsd">
   <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <DnsOverride>
            <id>57020</id>
        </DnsOverride>
    </data>
```

```
</ServiceResponse>
```

Delete DNS Override

/qps/rest/3.0/delete/was/dnsoverride

[POST]

Delete a DNS override that is in the user's scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access".

Input Parameters

Optional elements are used to retrieve DNS overrides to delete. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format.

Click here for available operators

Parameter	Description
id	(integer) The ID of the DNS override.
name	(text) The name given to the DNS override.
tags	Filter by tags applied.
tags.id	(integer) ID of the tag assigned to DNS override.
tags.name	(text) Tag name assigned to DNS override.
createdDate	(date) The date when the DNS override was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the DNS override was updated in WAS, in UTC date/time format.
owner.id	(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the DNS override.

owner.name	(text) Full name of the user who created the DNS override.
owner.username	(text) Username of the owner who created the DNS override. (like user_ab3).

Sample - Delete specific DNS override (POST)

API request

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

Request POST Data

```
<ServiceRequest>
    <filters>
        <Criteria field="id" operator="EQUALS">57020</Criteria>
    </filters>
    <data>
        <DnsOverride>
            <id>57220</id>
        </DnsOverride>
    </data>
</ServiceRequest>
```

XML 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/3.0
/was/dnsoverride.xsd">
   <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <DnsOverride>
            <id>57220</id>
        </DnsOverride>
    </data>
</ServiceResponse>
```

Reference: DNS Override

The <OptionProfile> element includes sub elements used to define an option profile. A reference of these elements is provided below. An asterisk * indicates a complex element.

Parameter	Description
id	(integer) The ID of the DNS override.
name	(text) The name given to the DNS override.
tags	Filter by tags applied.
tags.id	(integer) ID of the tag assigned to DNS override.
tags.name	(text) Tag name assigned to DNS override.
createdDate	(date) The date when the DNS override was created in WAS, in UTC date/time format.
updatedDate	(date) The date when the DNS override was updated in WAS, in UTC date/time format.
owner.id	(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the DNS override.
owner.name	(text) Full name of the user who created the DNS override.
owner.username	(text) Username of the owner who created the DNS override. (like user_ab3).
DnsMapping (keyword)	Use to configure the DNS override setting through API. You need to specify the hostname or FQDN and the corresponding IP address to be preferred for scanning. Example:

```
<set>
   <DnsMapping>
      <hostName>test</hostName>
       <ipAddress>2.3.4.5</ipAddress>
   </DnsMapping>
</set>
```

When you create a new DNS override, ensure:

-Name (Required): Name should be unique.

-Tags: The tag id should be valid and in scope of current user. Use only <Set> tag.

-Mappings (Required): Each mapping must have hostName and IpAddress in valid format. Use only <Set> tag.

-Comments: Only <Set> with 1 comment is allowed with maximum length 2048 characters.

When you update an DNS override, ensure:

-Name: In case of name update, the updated name should be unique.

-Id is required.

-At lease one of the following should be present other than id: Name, owner, tags, comments, mappings

-Tags: The <set> and <Add>/ <Removed> tags are mutually exclusive. Either use <set> or <Add> and <Removed>.

- Mappings: The <set> and <Add>/ <Removed> tags are mutually exclusive. Either use <set> or <Add> and <Removed>.

Burp

Import Burp Issues

/qps/rest/3.0/import/was/burp

[POST]

Imports Burp scan reports and store the findings discovered by the Burp Suite scanner with those discovered by WAS. You can import Burp reports to manage your Burp findings with WAS.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access” and WAS Permission “Import Burp Report”.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
webAppId	(integer)The web application ID. This element is assigned by the service and required for an update request.
purgeResults	(boolean) Set to false to indicate if all previous issues for the web application should be retained. By default, it is set to false. `Example: <purgeResults>false</purgeResults>`
closeUnreportedIssues	(boolean) Set to false to indicate if all previous issues for the web application should be marked as

	fixed and should not be reported. By default, it is set to false. `<closeUnreportedIssues>false</closeUnreportedIssues>`
fileName	(text) Name of the Burp XML file to be imported. If name is not specified, default format for the file name is API-ImportBurp-dd-mmm-yy hh:mm:ss

Sample - Import Burp Report

Let us import a burp report for web application with webAppID equal to 1052902. To import the Burp report, you need to specify the webAppID and then paste the contents of the burp results (XML) file in <burpXml> tag.

API request

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

Request POST data

```
<ServiceRequest>
<data>
<webAppId>1524084</webAppId>
<purgeResults>false</purgeResults>
<closeUnreportedIssues>false</closeUnreportedIssues>
<fileName>testBurpReportImport</fileName>
<burpXml><?xml version="1.0"?>
<!DOCTYPE issues [
<!ELEMENT issues (issue*)>
<!ATTLIST issues burpVersion CDATA "">
<!ATTLIST issues exportTime CDATA "">
<!ELEMENT issue (serialNumber, type, name, host, path, location,
severity, confidence, issueBackground?, remediationBackground?,
references?, vulnerabilityClassifications?, issueDetail?,
issueDetailItems?, remediationDetail?, requestresponse*,
collaboratorEvent*, infiltratorEvent*, staticAnalysis*,
dynamicAnalysis*)>
<!ELEMENT serialNumber (#PCDATA)>
```

```
<!ELEMENT type (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT host (#PCDATA)>
<!ATTLIST host ip CDATA "">
<!ELEMENT path (#PCDATA)>
<!ELEMENT location (#PCDATA)>
<!ELEMENT severity (#PCDATA)>
<!ELEMENT confidence (#PCDATA)>
<!ELEMENT issueBackground (#PCDATA)>
<!ELEMENT remediationBackground (#PCDATA)>
<!ELEMENT references (#PCDATA)>
<!ELEMENT vulnerabilityClassifications (#PCDATA)>
<!ELEMENT issueDetail (#PCDATA)>
<!ELEMENT issueDetailItems (issueDetailItem*)>
<!ELEMENT issueDetailItem (#PCDATA)>
<!ELEMENT remediationDetail (#PCDATA)>
<!ELEMENT requestresponse (request?, response?, responseRedirected?)>
<!ELEMENT request (#PCDATA)>
<!ATTLIST request method CDATA "">
<!ATTLIST request base64 (true|false) "false">
<!ELEMENT response (#PCDATA)>
<!ATTLIST response base64 (true|false) "false">
<!ELEMENT responseRedirected (#PCDATA)>
<!ELEMENT sender (#PCDATA)>
<!ELEMENT message (#PCDATA)>
<!ELEMENT conversation (#PCDATA)>
<!ELEMENT recipient (#PCDATA)>
<!ELEMENT recipients (recipient*)>
<!ELEMENT smtp (sender, recipients, message, conversation)>
<!ELEMENT collaboratorEvent (interactionType, originIp, time,
lookupType?, lookupHost?, requestresponse?, smtp?)>
<!ELEMENT interactionType (#PCDATA)>
<!ELEMENT originIp (#PCDATA)>
<!ELEMENT time (#PCDATA)>
<!ELEMENT lookupType (#PCDATA)>
<!ELEMENT lookupHost (#PCDATA)>
<!ELEMENT infiltratorEvent (parameterName, platform, signature,
stackTrace?, parameterValue?, collaboratorEvent)>
<!ELEMENT parameterName (#PCDATA)>
<!ELEMENT platform (#PCDATA)>
<!ELEMENT signature (#PCDATA)>
<!ELEMENT stackTrace (#PCDATA)>
<!ELEMENT parameterValue (#PCDATA)>
<!ELEMENT dynamicAnalysis (source, sink, sourceStackTrace,
sinkStackTrace, eventListenerStackTrace, sourceValue, sinkValue,
eventHandlerData, eventHandlerDataType, eventHandlerManipulatedData,
poc, origin, isOriginChecked, sourceElementId, sourceElementName,
eventFiredEventName, eventFiredElementId, eventFiredElementName,
eventFiredOuterHtml)>
<!ELEMENT staticAnalysis (source, sink, codeSnippets)>
<!ELEMENT source (#PCDATA)>
<!ELEMENT sink (#PCDATA)>
<!ELEMENT sourceStackTrace (#PCDATA)>
<!ELEMENT sinkStackTrace (#PCDATA)>
<!ELEMENT eventListenerStackTrace (#PCDATA)>
<!ELEMENT sourceValue (#PCDATA)>
<!ELEMENT sinkValue (#PCDATA)>
<!ELEMENT eventHandlerData (#PCDATA)>
<!ELEMENT eventHandlerDataType (#PCDATA)>
<!ELEMENT sourceElementId (#PCDATA)>
<!ELEMENT sourceElementName (#PCDATA)>
<!ELEMENT eventFiredEventName (#PCDATA)>
<!ELEMENT eventFiredElementId (#PCDATA)>
<!ELEMENT eventFiredElementName (#PCDATA)>
<!ELEMENT eventFiredOuterHtml (#PCDATA)>
<!ELEMENT eventHandlerManipulatedData (#PCDATA)>
<!ELEMENT poc (#PCDATA)>
<!ELEMENT origin (#PCDATA)>
<!ELEMENT isOriginChecked (#PCDATA)>
<!ELEMENT codeSnippets (codeSnippet*)>
<!ELEMENT codeSnippet (#PCDATA)>
]>
<issues burpVersion="2.0.20beta" exportTime="Wed May 29 08:45:42 CDT
2019">
  <issue>
    <serialNumber>5018346890832155648</serialNumber>
    <type>16777728</type>
    <name><![CDATA[Unencrypted communications]]></name>
    <host ip="172.217.164.116">http://google-
gruyere.appspot.com</host>
    <path><![CDATA[/]]></path>
    <location><![CDATA[/]]></location>
    <severity>Low</severity>
    <confidence>Certain</confidence>
    <issueBackground><![CDATA[<p>The application allows users to
connect to it over unencrypted connections.  An attacker suitably
positioned to view a legitimate user's network traffic could record
and monitor their interactions with the application and obtain any
information the user supplies. Furthermore, an attacker able to modify
traffic could use the application as a platform for attacks against
```

```
its users and third-party websites. Unencrypted connections have been
exploited by ISPs and governments to track users, and to inject
adverts and malicious JavaScript. Due to these concerns, web browser
vendors are planning to visually flag unencrypted connections as
hazardous.</p>
<p>
To exploit this vulnerability, an attacker must be suitably positioned
to eavesdrop on the victim's network traffic. This scenario typically
occurs when a client communicates with the server over an insecure
connection such as public Wi-Fi, or a corporate or home network that
is shared with a compromised computer. Common defenses such as
switched networks are not sufficient to prevent this. An attacker
situated in the user's ISP or the application's hosting infrastructure
could also perform this attack. Note that an advanced adversary could
potentially target any connection made over the Internet's core
infrastructure.
</p>
<p>Please note that using a mixture of encrypted and unencrypted
communications is an ineffective defense against active attackers,
because they can easily remove references to encrypted resources when
these references are transmitted over an unencrypted
connection.</p>]]></issueBackground>
    <remediationBackground><![CDATA[<p>Applications should use
transport-level encryption (SSL/TLS) to protect all communications
passing between the client and the server. The Strict-Transport-
Security HTTP header should be used to ensure that clients refuse to
access the server over an insecure
connection.</p>]]></remediationBackground>
    <references><![CDATA[<ul>
<li><a href="https://www.chromium.org/Home/chromium-security/marking-
http-as-non-secure">Marking HTTP as non-secure</a></li>
<li><a
href="https://wiki.mozilla.org/Security/Server_Side_TLS">Configuring
Server-Side SSL/TLS</a></li>
<li><a href="https://developer.mozilla.org/en-
US/docs/Web/Security/HTTP_strict_transport_security">HTTP Strict
Transport Security</a></li>
</ul>]]></references>
    <vulnerabilityClassifications><![CDATA[<ul>
<li><a href="https://cwe.mitre.org/data/definitions/326.html">CWE-326:
Inadequate Encryption Strength</a></li>
</ul>]]></vulnerabilityClassifications>
  </issue>
  <issue>
    <serialNumber>5761124851012705280</serialNumber>
    <type>2097920</type>
    <name><![CDATA[Cross-site scripting (reflected)]]></name>
    <host ip="172.217.164.116">http://google-
gruyere.appspot.com</host>
    <path><![CDATA[/922324844025/login]]></path>
    <location><![CDATA[/922324844025/login [URL path
filename]]]></location>
    <severity>High</severity>
    <confidence>Certain</confidence>
    <issueBackground><![CDATA[<p>Reflected cross-site scripting
vulnerabilities arise when data is copied from a request and echoed
into the application's immediate response in an unsafe way. An
attacker can use the vulnerability to construct a request that, if
issued by another application user, will cause JavaScript code
supplied by the attacker to execute within the user's browser in the
context of that user's session with the application.</p>
<p>The attacker-supplied code can perform a wide variety of actions,
such as stealing the victim's session token or login credentials,
performing arbitrary actions on the victim's behalf, and logging their
keystrokes.</p>
<p>Users can be induced to issue the attacker's crafted request in
various ways. For example, the attacker can send a victim a link
containing a malicious URL in an email or instant message. They can
submit the link to popular web sites that allow content authoring, for
example in blog comments. And they can create an innocuous looking web
site that causes anyone viewing it to make arbitrary cross-domain
requests to the vulnerable application (using either the GET or the
POST method).</p>
<p>The security impact of cross-site scripting vulnerabilities is
dependent upon the nature of the vulnerable application, the kinds of
data and functionality that it contains, and the other applications
that belong to the same domain and organization. If the application is
used only to display non-sensitive public content, with no
authentication or access control functionality, then a cross-site
scripting flaw may be considered low risk. However, if the same
application resides on a domain that can access cookies for other more
security-critical applications, then the vulnerability could be used
to attack those other applications, and so may be considered high
risk. Similarly, if the organization that owns the application is a
likely target for phishing attacks, then the vulnerability could be
leveraged to lend credibility to such attacks, by injecting Trojan
functionality into the vulnerable application and exploiting users'
trust in the organization in order to capture credentials for other
applications that it owns. In many kinds of application, such as those
```

```
providing online banking functionality, cross-site scripting should
always be considered high risk. </p>]]></issueBackground>
    <remediationBackground><![CDATA[<p>In most situations where user-
controllable data is copied into application responses, cross-site
scripting
  attacks can be prevented using two layers of defenses:</p>
<ul>
  <li>Input should be validated as strictly as possible on arrival,
given the kind of content that
it is expected to contain. For example, personal names should consist
of alphabetical
and a small range of typographical characters, and be relatively
short; a year of birth
should consist of exactly four numerals; email addresses should match
a well-defined
regular expression. Input which fails the validation should be
rejected, not sanitized.</li>
<li>User input should be HTML-encoded at any point where it is copied
into
application responses. All HTML metacharacters, including &lt; &gt; "
' and =, should be
replaced with the corresponding HTML entities (&amp;lt; &amp;gt;
etc).</li></ul>
<p>In cases where the application's functionality allows users to
author content using
  a restricted subset of HTML tags and attributes (for example, blog
comments which
  allow limited formatting and linking), it is necessary to parse the
supplied HTML to
  validate that it does not use any dangerous syntax; this is a non-
trivial task.</p>]]></remediationBackground>
    <references><![CDATA[<ul><li><a
href="https://support.portswigger.net/customer/portal/articles/1965737
-Methodology_XSS.html">Using Burp to Find XSS
issues</a></li></ul>]]></references>
    <vulnerabilityClassifications><![CDATA[<ul>
<li><a href="https://cwe.mitre.org/data/definitions/79.html">CWE-79:
Improper Neutralization of Input During Web Page Generation ('Cross-
site Scripting')</a></li>
<li><a href="https://cwe.mitre.org/data/definitions/80.html">CWE-80:
Improper Neutralization of Script-Related HTML Tags in a Web Page
(Basic XSS)</a></li>
<li><a href="https://cwe.mitre.org/data/definitions/116.html">CWE-116:
Improper Encoding or Escaping of Output</a></li>
```

```
<li><a href="https://cwe.mitre.org/data/definitions/159.html">CWE-159:
Failure to Sanitize Special Element</a></li>
</ul>]]></vulnerabilityClassifications>
    <issueDetail><![CDATA[The value of the URL path filename is copied
into the HTML document as plain text between tags. The payload
<b>bpi9f&lt;script&gt;alert(1)&lt;/script&gt;j4wjy</b> was submitted
in the URL path filename. This input was echoed unmodified in the
application's response.<br><br>This proof-of-concept attack
demonstrates that it is possible to inject arbitrary JavaScript into
the application's response.]]></issueDetail>
    <requestresponse>
      <request method="GET"
base64="true"><![CDATA[R0VUIC85MjIzMjQ4NDQwMjUvbG9naW5icGk5ZiUzY3Njcml
wdCUzZWFsZXJ0KDEpJTNjL3NjcmlwdCUzZWo0d2p5P3VpZD1hYWFhJnB3PWJiYmIgSFRUU
C8xLjENCkhvc3Q6IGdvb2dsZS1ncnV5ZXJlLmFwcHNwb3QuY29tDQpVcGdyYWRlLUluc2V
jdXJlLVJlcXVlc3RzOiAxDQpVc2VyLUFnZW50OiBNb3ppbGxhLzUuMCAoV2luZG93cyBOV
CAxMC4wOyBXaW42NDsgeDY0KSBBcHBsZVdlYktpdC81MzcuMzYgKEtIVE1MLCBsaWtlIEd
lY2tvKSBDaHJvbWUvNzQuMC4zNzI5LjE1NyBTYWZhcmkvNTM3LjM2DQpBY2NlcHQ6IHRle
HQvaHRtbCxhcHBsaWNhdGlvbi94aHRtbCt4bWwsYXBwbGljYXRpb24veG1sO3E9MC45LGl
tYWdlL3dlYnAsaW1hZ2UvYXBuZywqLyo7cT0wLjgsYXBwbGljYXRpb24vc2lnbmVkLWV4Y
2hhbmdlO3Y9YjMNClJlZmVyZXI6IGh0dHA6Ly9nb29nbGUtZ3J1eWVyZS5hcHBzcG90LmN
vbS85MjIzMjQ4NDQwMjUvbG9naW4NCkFjY2VwdC1FbmNvZGluZzogZ3ppcCwgZGVmbGF0Z
Q0KQWNjZXB0LUxhbmd1YWdlOiBlbi1VUyxlbjtxPTAuOQ0KQ29va2llOiBHUlVZRVJFPQ0
KQ29ubmVjdGlvbjogY2xvc2UNCg0K]]></request>
      <response
base64="true"><![CDATA[SFRUUC8xLjEgMjAwIE9LDQpDYWNoZS1Db250cm9sOiBuby1
jYWNoZQ0KQ29udGVudC10eXBlOiB0ZXh0L2h0bWwNClByYWdtYTogbm8tY2FjaGUNClgtW
FNTLVByb3RlY3Rpb246IDANClgtQ2xvdWQtVHJhY2UtQ29udGV4dDogMThlMjRkZDkyZTY
wYTNlMTQ0NWJkMTQxNmJmYTAxMGYNClZhcnk6IEFjY2VwdC1FbmNvZGluZw0KRGF0ZTogV
2VkLCAyOSBNYXkgMjAxOSAxMzo0MTowNiBHTVQNClNlcnZlcjogR29vZ2xlIEZyb250ZW5
kDQpDb250ZW50LUxlbmd0aDogMjE4NQ0KQ29ubmVjdGlvbjogY2xvc2UNCg0KPCFET0NUW
VBFIEhUTUwgUFVCTElDICItLy9XM0MvL0RURCBIVE1MIDQuMDEgVHJhbnNpdGlvbmFsLy9
FTiI+CjwhLS0gQ29weXJpZ2h0IDIwMTcgR29vZ2xlIEluYy4gLS0+CjxodG1sPgo8aGVhZ
D4KPHRpdGxlPkdydXllcmU6IEVycm9yPC90aXRsZT4KPHN0eWxlPgovKiBDb3B5cmlnaHQ
gMjAxNyBHb29nbGUgSW5jLiAqLwoKYm9keSwgaHRtbCwgdGQsIHNwYW4sIGRpdiwgaW5wd
XQsIHRleHRhcmVhIHsKICBmb250LWZhbWlseTogc2Fucy1zZXJpZjsKICBmb250LXNpemU
6IDE0cHQ7Cn0KCmJvZHkgewogIGJhY2tncm91bmQ6IHVybCgnY2hlZXNlLnBuZycpIHRvc
CBjZW50ZXIgcmVwZWF0OwogIHRleHQtYWxpZ246IGNlbnRlcjsKICBvcGFjaXR5OiAwLjg
wOwp9CgpoMiB7CiAgdGV4dC1hbGlnbjogY2VudGVyOwogIGZvbnQtc2l6ZTogMzBwdDsKI
CBmb250LXdlaWdodDogYm9sZDsKfQoKdGQgewogIHZlcnRpY2FsLWFsaWduOiB0b3A7CiA
gcGFkZGluZzogNXB4Owp9CgphLCBhOmhvdmVyIHsKICB0ZXh0LWRlY29yYXRpb246IHVuZ
GVybGluZTsKICBjb2xvcjogIzAwMDBiYjsKfQoKYTp2aXNpdGVkIHsKICBjb2xvcjogI2J
iMDAwMDsKfQoKYS5idXR0b246dmlzaXRlZCB7CiAgY29sb3I6ICMwMDAwYmI7Cn0KCi5jb
250ZW50IHsKICB0ZXh0LWFsaWduOiBsZWZ0OwogIG1hcmdpbi1sZWZ0OiBhdXRvOwogIG1
hcmdpbi1yaWdodDogYXV0bzsKICB3aWR0aDogOTAlOwogIGJhY2tncm91bmQ6ICNmZmZmY
```

```
2M7CiAgcGFkZGluZzogMjBweDsKICBib3JkZXI6IDNweCBzb2xpZCAjZmZiMTQ5Owp9Cgo
ubWVudSB7CiAgdGV4dC1hbGlnbjogbGVmdDsKICBwYWRkaW5nOiAxMHB4IDIwcHggMzVwe
CAyMHB4OwogIG1hcmdpbi1sZWZ0OiBhdXRvOwogIG1hcmdpbi1yaWdodDogYXV0bzsKICB
tYXJnaW4tdG9wOiAyMHB4OwogIHdpZHRoOiA5MCU7CiAgYmFja2dyb3VuZDogI2ZmZmZjY
zsKICBib3JkZXI6IDNweCBzb2xpZCAjZmZiMTQ5Owp9CgoubWVudS11c2VyIHsKICBjb2x
vcjogIzAwMDAwMDsKICBmb250LXdlaWdodDogYm9sZDsKfQoKI21lbnUtbGVmdCB7CiAgZ
mxvYXQ6IGxlZnQ7Cn0KCiNtZW51LWxlZnQgYSwgI21lbnUtbGVmdCBhOmhvdmVyLCAjbWV
udS1sZWZ0IGE6dmlzaXRlZCB7CiAgY29sb3I6ICMwMDAwMDA7Cn0KCiNtZW51LXJpZ2h0I
HsKICBmbG9hdDogcmlnaHQ7Cn0KCiNtZW51LXJpZ2h0IGEsICNtZW51LXJpZ2h0IGE6aG9
2ZXIsICNtZW51LXJpZ2h0IGE6dmlzaXRlZCB7CiAgY29sb3I6ICMwMDAwMDA7Cn0KCi5tZ
XNzYWdlIHsKICB3aWR0aDogNTAlOwogIGNvbG9yOiAjZmYwMDAwOwogIGJhY2tncm91bmQ
6ICNmZmRkZGQ7CiAgYm9yZGVyOiAycHggc29saWQgI2ZmMDAwMDsKICBib3JkZXItcmFka
XVzOiAxZW07CiAgLW1vei1ib3JkZXItcmFkaXVzOiAxZW07CiAgcGFkZGluZzogMTBweDs
KICBmb250LXdlaWdodDogYm9sZDsKICB0ZXh0LWFsaWduOiBjZW50ZXI7CiAgbWFyZ2luO
iBhdXRvOwogIG1hcmdpbi10b3A6IDIwcHg7CiAgbWFyZ2luLWJvdHRvbTogMjBweDsKfQo
KaW5wdXQsIHRleHRhcmVhIHsKICBiYWNrZ3JvdW5kLWNvbG9yOiAjZmZmZmZmOwp9Cgouc
mVmcmVzaCB7CiAgZmxvYXQ6IGNlbnRlcjsKICB3aWR0aDogOTAlOwogIHRleHQtYWxpZ24
6IHJpZ2h0OwogIG1hcmdpbjogYXV0bzsKICBwYWRkaW5nLXRvcDogMDsKICBwYWRkaW5nL
WJvdHRvbTogMnB0OwogIG1hcmdpbi10b3A6IDA7CiAgbWFyZ2luLWJvdHRvbTogMDsKfQo
KLmgyLXdpdGgtcmVmcmVzaCB7CiAgbWFyZ2luLWJvdHRvbTogMDsKfQoKPC9zdHlsZT4KC
jwvaGVhZD4KCjxib2R5PgoKPGRpdiBjbGFzcz0nbWVudSc+CiAgPHNwYW4gaWQ9J21lbnU
tbGVmdCc+CiAgICA8YSBocmVmPScvOTIyMzI0ODQ0MDI1Lyc+SG9tZTwvYT4KICAgICAgC
iAgPC9zcGFuPgogIDxzcGFuIGlkPSdtZW51LXJpZ2h0Jz4KICAgICAgCiAgICAgIAogICA
gICA8YSBocmVmPScvOTIyMzI0ODQ0MDI1L2xvZ2luJz5TaWduIGluPC9hPgogICAgICB8I
DxhIGhyZWY9Jy85MjIzMjQ4NDQwMjUvbmV3YWNjb3VudC5ndGwnPlNpZ24gdXA8L2E+CiA
gICAgIAogIDwvc3Bhbj4KPC9kaXY+CgoKCjxkaXYgY2xhc3M9J21lc3NhZ2UnPkludmFsa
WQgcmVxdWVzdDogL2xvZ2luYnBpOWY8c2NyaXB0PmFsZXJ0KDEpPC9zY3JpcHQ+ajR3ank
8L2Rpdj4KCgo8L2JvZHk+Cgo8L2h0bWw+Cg==]]></response>
      <responseRedirected>false</responseRedirected>
    </requestresponse>
  </issue>
  <issue>
    <serialNumber>7919395047422736384</serialNumber>
    <type>5244416</type>
    <name><![CDATA[Cookie without HttpOnly flag set]]></name>
    <host ip="172.217.164.116">http://google-
gruyere.appspot.com</host>
    <path><![CDATA[/922324844025/saveprofile]]></path>
    <location><![CDATA[/922324844025/saveprofile]]></location>
    <severity>Information</severity>
    <confidence>Certain</confidence>
    <issueBackground><![CDATA[<p>If the HttpOnly attribute is set on a
cookie, then the cookie's value cannot be read or set by client-side
JavaScript. This measure makes certain client-side attacks, such as
cross-site scripting, slightly harder to exploit by preventing them
```

```
from trivially capturing the cookie's value via an injected
script.</p>]]></issueBackground>
    <remediationBackground><![CDATA[<p>There is usually no good reason
not to set the HttpOnly flag on all cookies. Unless you specifically
require legitimate client-side scripts within your application to read
or set a cookie's value, you should set the HttpOnly flag by including
this attribute within the relevant Set-cookie directive.</p>
<p>You should be aware that the restrictions imposed by the HttpOnly
flag can potentially be circumvented in some circumstances, and that
numerous other serious attacks can be delivered by client-side script
injection, aside from simple cookie stealing.
</p>]]></remediationBackground>
    <references><![CDATA[<ul>
<li><a href='https://www.owasp.org/index.php/HttpOnly'>Configuring
HttpOnly</a></li>
</ul>]]></references>
    <vulnerabilityClassifications><![CDATA[<ul>
<li><a href="https://cwe.mitre.org/data/definitions/16.html">CWE-16:
Configuration</a></li>
</ul>]]></vulnerabilityClassifications>
    <issueDetail><![CDATA[The following cookie was issued by the
application and does not have the HttpOnly flag
set:<ul><li>GRUYERE</li></ul>The cookie does not appear to contain a
session token, which may reduce the risk associated with this issue.
You should review the contents of the cookie to determine its
function.]]></issueDetail>
    <issueDetailItems>
      <issueDetailItem><![CDATA[Other: GRUYERE]]></issueDetailItem>
    </issueDetailItems>
    <requestresponse>
      <request method="GET"
base64="true"><![CDATA[R0VUIC85MjIzMjQ4NDQwMjUvc2F2ZXByb2ZpbGU/YWN0aW9
uPW5ldyZ1aWQ9YWFhYSZwdz1iYmJiYiZpc19hdXRob3I9VHJ1ZSBIVFRQLzEuMQ0KSG9zd
DogZ29vZ2xlLWdydXllcmUuYXBwc3BvdC5jb20NClVwZ3JhZGUtSW5zZWN1cmUtUmVxdWV
zdHM6IDENClVzZXItQWdlbnQ6IE1vemlsbGEvNS4wIChXaW5kb3dzIE5UIDEwLjA7IFdpb
jY0OyB4NjQpIEFwcGxlV2ViS2l0LzUzNy4zNiAoS0hUTUwsIGxpa2UgR2Vja28pIENocm9
tZS83NC4wLjM3MjkuMTU3IFNhZmFyaS81MzcuMzYNCkFjY2VwdDogdGV4dC9odG1sLGFwc
GxpY2F0aW9uL3hodG1sK3htbCxhcHBsaWNhdGlvbi94bWw7cT0wLjksaW1hZ2Uvd2VicCx
pbWFnZS9hcG5nLCovKjtxPTAuOCxhcHBsaWNhdGlvbi9zaWduZWQtZXhjaGFuZ2U7dj1iM
w0KUmVmZXJlcjogaHR0cDovL2dvb2dsZS1ncnV5ZXJlLmFwcHNwb3QuY29tLzkyMjMyNDg
0NDAyNS9uZXdhY2NvdW50Lmd0bA0KQWNjZXB0LUVuY29kaW5nOiBnemlwLCBkZWZsYXRlD
QpBY2NlcHQtTGFuZ3VhZ2U6IGVuLVVTLGVuO3E9MC45DQpDb25uZWN0aW9uOiBjbG9zZQ0
KDQo=]]></request>
      <response
base64="true"><![CDATA[SFRUUC8xLjEgMjAwIE9LDQpDYWNoZS1Db250cm9sOiBuby1
```

```
jYWNoZQ0KQ29udGVudC10eXBlOiB0ZXh0L2h0bWwNClByYWdtYTogbm8tY2FjaGUNClNld
C1Db29raWU6IEdSVVlFUkU9ODQ3Nzc1MzB8YWFhYXx8YXV0aG9yOyBwYXRoPS85MjIzMjQ
4NDQwMjUNClgtWFNTLVByb3RlY3Rpb246IDANClgtQ2xvdWQtVHJhY2UtQ29udGV4dDogY
2YwYWVmNDQyNzIwNGRjZjJhMGE5OTEyOTE0YTIyMWQNClZhcnk6IEFjY2VwdC1FbmNvZGl
uZw0KRGF0ZTogV2VkLCAyOSBNYXkgMjAxOSAxMzozNjowMiBHTVQNClNlcnZlcjogR29vZ
2xlIEZyb250ZW5kDQpDb250ZW50LUxlbmd0aDogMjE0Mw0KRXhwaXJlczogV2VkLCAyOSB
NYXkgMjAxOSAxMzozNjowMiBHTVQNCkNvbm5lY3Rpb246IGNsb3NlDQoNCjwhRE9DVFlQR
SBIVE1MIFBVQkxJQyAiLS8vVzNDLy9EVEQgSFRNTCA0LjAxIFRyYW5zaXRpb25hbC8vRU4
iPgo8IS0tIENvcHlyaWdodCAyMDE3IEdvb2dsZSBJbmMuIC0tPgo8aHRtbD4KPGhlYWQ+C
jx0aXRsZT5HcnV5ZXJlOiBFcnJvcjwvdGl0bGU+CjxzdHlsZT4KLyogQ29weXJpZ2h0IDI
wMTcgR29vZ2xlIEluYy4gKi8KCmJvZHksIGh0bWwsIHRkLCBzcGFuLCBkaXYsIGlucHV0L
CB0ZXh0YXJlYSB7CiAgZm9udC1mYW1pbHk6IHNhbnMtc2VyaWY7CiAgZm9udC1zaXplOiA
xNHB0Owp9Cgpib2R5IHsKICBiYWNrZ3JvdW5kOiB1cmwoJ2NoZWVzZS5wbmcnKSB0b3AgY
2VudGVyIHJlcGVhdDsKICB0ZXh0LWFsaWduOiBjZW50ZXI7CiAgb3BhY2l0eTogMC44MDs
KfQoKaDIgewogIHRleHQtYWxpZ246IGNlbnRlcjsKICBmb250LXNpemU6IDMwcHQ7CiAgZ
m9udC13ZWlnaHQ6IGJvbGQ7Cn0KCnRkIHsKICB2ZXJ0aWNhbC1hbGlnbjogdG9wOwogIHB
hZGRpbmc6IDVweDsKfQoKYSwgYTpob3ZlciB7CiAgdGV4dC1kZWNvcmF0aW9uOiB1bmRlc
mxpbmU7CiAgY29sb3I6ICMwMDAwYmI7Cn0KCmE6dmlzaXRlZCB7CiAgY29sb3I6ICNiYjA
wMDA7Cn0KCmEuYnV0dG9uOnZpc2l0ZWQgewogIGNvbG9yOiAjMDAwMGJiOwp9CgouY29ud
GVudCB7CiAgdGV4dC1hbGlnbjogbGVmdDsKICBtYXJnaW4tbGVmdDogYXV0bzsKICBtYXJ
naW4tcmlnaHQ6IGF1dG87CiAgd2lkdGg6IDkwJTsKICBiYWNrZ3JvdW5kOiAjZmZmZmNjO
wogIHBhZGRpbmc6IDIwcHg7CiAgYm9yZGVyOiAzcHggc29saWQgI2ZmYjE0OTsKfQoKLm1
lbnUgewogIHRleHQtYWxpZ246IGxlZnQ7CiAgcGFkZGluZzogMTBweCAyMHB4IDM1cHggM
jBweDsKICBtYXJnaW4tbGVmdDogYXV0bzsKICBtYXJnaW4tcmlnaHQ6IGF1dG87CiAgbWF
yZ2luLXRvcDogMjBweDsKICB3aWR0aDogOTAlOwogIGJhY2tncm91bmQ6ICNmZmZmY2M7C
iAgYm9yZGVyOiAzcHggc29saWQgI2ZmYjE0OTsKfQoKLm1lbnUtdXNlciB7CiAgY29sb3I
6ICMwMDAwMDA7CiAgZm9udC13ZWlnaHQ6IGJvbGQ7Cn0KCiNtZW51LWxlZnQgewogIGZsb
2F0OiBsZWZ0Owp9CgojbWVudS1sZWZ0IGEsICNtZW51LWxlZnQgYTpob3ZlciwgI21lbnU
tbGVmdCBhOnZpc2l0ZWQgewogIGNvbG9yOiAjMDAwMDAwOwp9CgojbWVudS1yaWdodCB7C
iAgZmxvYXQ6IHJpZ2h0Owp9CgojbWVudS1yaWdodCBhLCAjbWVudS1yaWdodCBhOmhvdmV
yLCAjbWVudS1yaWdodCBhOnZpc2l0ZWQgewogIGNvbG9yOiAjMDAwMDAwOwp9CgoubWVzc
2FnZSB7CiAgd2lkdGg6IDUwJTsKICBjb2xvcjogI2ZmMDAwMDsKICBiYWNrZ3JvdW5kOiA
jZmZkZGRkOwogIGJvcmRlcjogMnB4IHNvbGlkICNmZjAwMDA7CiAgYm9yZGVyLXJhZGl1c
zogMWVtOwogIC1tb3otYm9yZGVyLXJhZGl1czogMWVtOwogIHBhZGRpbmc6IDEwcHg7CiA
gZm9udC13ZWlnaHQ6IGJvbGQ7CiAgdGV4dC1hbGlnbjogY2VudGVyOwogIG1hcmdpbjogY
XV0bzsKICBtYXJnaW4tdG9wOiAyMHB4OwogIG1hcmdpbi1ib3R0b206IDIwcHg7Cn0KCml
ucHV0LCB0ZXh0YXJlYSB7CiAgYmFja2dyb3VuZC1jb2xvcjogI2ZmZmZmZjsKfQoKLnJlZ
nJlc2ggewogIGZsb2F0OiBjZW50ZXI7CiAgd2lkdGg6IDkwJTsKICB0ZXh0LWFsaWduOiB
yaWdodDsKICBtYXJnaW46IGF1dG87CiAgcGFkZGluZy10b3A6IDA7CiAgcGFkZGluZy1ib
3R0b206IDJwdDsKICBtYXJnaW4tdG9wOiAwOwogIG1hcmdpbi1ib3R0b206IDA7Cn0KCi5
oMi13aXRoLXJlZnJlc2ggewogIG1hcmdpbi1ib3R0b206IDA7Cn0KCjwvc3R5bGU+Cgo8L
2hlYWQ+Cgo8Ym9keT4KCjxkaXYgY2xhc3M9J21lbnUnPgogIDxzcGFuIGlkPSdtZW51LWx
lZnQnPgogICAgPGEgaHJlZj0nLzkyMjMyNDg0NDAyNS8nPkhvbWU8L2E+CiAgICAgIAogI
Dwvc3Bhbj4KICA8c3BhbiBpZD0nbWVudS1yaWdodCc+CiAgICAgIAogICAgICAKICAgICA
gPGEgaHJlZj0nLzkyMjMyNDg0NDAyNS9sb2dpbic+U2lnbiBpbjwvYT4KICAgICAgfCA8Y
```

```
SBocmVmPScvOTIyMzI0ODQ0MDI1L25ld2FjY291bnQuZ3RsJz5TaWduIHVwPC9hPgogICA
gICAKICA8L3NwYW4+CjwvZGl2PgoKCgo8ZGl2IGNsYXNzPSdtZXNzYWdlJz5BY2NvdW50I
GNyZWF0ZWQuPC9kaXY+CgoKPC9ib2R5PgoKPC9odG1sPgo=]]></response>
      <responseRedirected>false</responseRedirected>
    </requestresponse>
  </issue>
</issues>
</burpXml>
</data>
</ServiceRequest>
```

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/3.
0/was/burp.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Burp>
            <id>145201</id>
            <webApp>
                <id>1524084</id>
                <name>
                    <![CDATA[demoap15webapp]]>
                </name>
                <url>
                    <![CDATA[http://10.11.72.37]]>
                </url>
            </webApp>
            <issuesCount>3</issuesCount>
            <issues burpVersion="2.0.20beta" exportTime="Wed May 29
13:45:42 UTC 2019">
                <issue>
                    <id>174201</id>
                    <serialNumber>5018346890832155648</serialNumber>
                </issue>
                <issue>
                    <id>174202</id>
                    <serialNumber>5761124851012705280</serialNumber>
                </issue>
                <issue>
                    <id>174203</id>
                    <serialNumber>7919395047422736384</serialNumber>
                </issue>
            </issues>
            <fileName>testBurpReportImport</fileName>
            <errorRecords>
                <count>0</count>
            </errorRecords>
        </Burp>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/burp.xsd

OWASP ZAP

Import OWASP ZAP Findings

/qps/rest/3.0/import/was/owaspzap/

[POST]

Support for importing OWASP ZAP reports and save the findings discovered by OWASP ZAP tool with the findings discovered by WAS.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission "API Access", WAS Permission "Access OWASP ZAP Report" and "Import OWASP ZAP Report" .

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
webAppId	(integer)The web application ID. This element is assigned by the service and required for an update request.
purgeResults	(boolean) Set to false to indicate if all previous issues for the web application should be retained. By default, it is set to false. `Example: <purgeResults>false</purgeResults>`
closeUnreportedIssues	(boolean) Set to false to indicate if all previous issues for the web application should be marked as

	fixed and should not be reported. By default, it is set to false. `<closeUnreportedIssues>false</closeUnreportedIssues>`
fileName	(text) Name of the OWAS zap XML file to be imported. If name is not specified, default format for the file name is API-ImportOwaspZap-dd-mmm-yy hh:mm:ss

Sample - Import OWASP ZAP Findings

Let us import a OWASP ZAP reports for web application with webAppID equal to 29120395.

API request

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" -
data-binary
@"https://<QualysBaseURL>/qps/rest/3.0/import/was/owaspzap" <file.xml>
```

Request POST data

```
<ServiceRequest>
<data>
    <webAppId>29120395</webAppId>
    <purgeResults>false</purgeResults>
    <closeUnreportedIssues>false</closeUnreportedIssues>
    <fileName>testOwaspFile</fileName>
    <owaspZapXml>
    <OWASPZAPReport programName="OWASP ZAP" version="Dev Build"
generated="Thu, 17 Nov 2022 11:03:08">
    <site name="https://www.googletagservices.com"
host="www.googletagservices.com" port="443" ssl="true">
    <alerts>
        <alertitem>
        <pluginid>10035</pluginid>
        <alertRef>10035</alertRef>
        <alert>Strict-Transport-Security Header NotSet</alert>
        <name>Strict-Transport-Security Header Not Set</name>
        <riskcode>1</riskcode>
        <confidence>3</confidence>
        <riskdesc>Low (High)</riskdesc>
        <confidencedesc>High</confidencedesc>
        <desc>HTTP Strict Transport Security (HSTS) is a web security
policy mechanism whereby a web server declares that complying user
agents (such as a web browser) are to interact with it using only
secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an
IETF standards track protocol and is specified in RFC 6797.</desc>
        <instances>
        <instance>
        <uri>https://www.googletagservices.com/tag/js/gpt.js</uri>
        <method>GET</method>
        <param></param>
        <attack></attack>
        <evidence></evidence>
        <requestheader>GET
https://www.googletagservices.com/tag/js/gpt.js HTTP/1.1 Host:
www.googletagservices.com Connection: keep-alive sec-ch-ua:
&quot;Google Chrome&quot;;v=&quot;107&quot;,
&quot;Chromium&quot;;v=&quot;107&quot;,
&quot;Not=A?Brand&quot;;v=&quot;24&quot; sec-ch-ua-mobile: ?0 User-
Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 sec-ch-ua-platform:
&quot;Windows&quot; Accept: */* Sec-Fetch-Site: cross-site Sec-Fetch-
Mode: no-cors Sec-Fetch-Dest: script Referer: https://jsonlint.com/
Accept-Language: en-US,en;q=0.9 </requestheader>
        <requestbody></requestbody>
        <responseheader>HTTP/1.1 200 OK Vary: Accept-Encoding Content-
Type: text/javascript Cross-Origin-Resource-Policy: cross-origin
Cross-Origin-Opener-Policy-Report-Only: same-origin; report-
to=&quot;adsgpt-scs&quot; Report-To: {&quot;group&quot;:&quot;ads-
gptscs&quot;,&quot;max_age&quot;:2592000,&quot;endpoints&quot;:[{&quot
;url&q uot;:&quot;https://csp.withgoogle.com/csp/report-to/ads-gpt-
scs&quot;}]} Timing-Allow-Origin: * Content-Length: 80512 Date: Thu,
17 Nov 2022 05:20:21 GMT Expires: Thu, 17 Nov 2022 05:20:21 GMT Cache-
Control: private, max-age=900, stale-while-revalidate=3600 ETag:
&quot;1394 / 793 of 1000 / last-modified: 1668639967&quot; X-Content-
Type-Options: nosniff Server: sffe X-XSS-Protection: 0 Alt-Svc:
h3=&quot;:443&quot;; ma=2592000,h3-29=&quot;:443&quot;; ma=2592000,h3-
Q050=&quot;:443&quot;; ma=2592000,h3-Q046=&quot;:443&quot;;
ma=2592000,h3-Q043=&quot;:443&quot;; ma=2592000,quic=&quot;:443&quot;;
ma=2592000; v=&quot;46,43&quot; </responseheader>
        <responsebody>(function(E)) </responsebody>
        </instance>
        </instances>
        <count>1</count>
        <solution>Ensure that your web server, application server,
load balancer, etc. is configured to enforce Strict-Transport-
Security.</solution>
        <otherinfo></otherinfo>
        <reference>https://cheatsheetseries.owasp.org/cheatsheets/HTTP
_Strict_Tra nsport_Security_Cheat_Sheet.html https://owasp.org/www-
community/Security_Headers
http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
http://caniuse.com/stricttransportsecurity
http://tools.ietf.org/html/rfc6797</reference>
        <cweid>319</cweid>
        <wascid>15</wascid>
        <sourceid>8</sourceid>
        <tags>
        <tag>
        <tag>OWASP_2021_A05</tag>
        <link>https://owasp.org/Top10/A05_2021Security_Misconfiguratio
n/</link>
        </tag>
        <tag>
        <tag>OWASP_2017_A06</tag>
        <link>https://owasp.org/www-projecttop-ten/2017/A6_2017-
Security_Misconfiguration.html</link>
        </tag>
        </tags>
        </alertitem>
        <alertitem>
        <pluginid>10027</pluginid>
        <alertRef>10027</alertRef>
        <alert>Information Disclosure - Suspicious Comments</alert>
        <name>Information Disclosure - Suspicious Comments</name>
        <riskcode>0</riskcode>
        <confidence>1</confidence>
        <riskdesc>Informational (Low)</riskdesc>
        <confidencedesc>Low</confidencedesc>
        <desc>The response appears to contain suspicious comments
which may help an attacker. Note: Matches made within script blocks or
files are against the entire content not only comments.</desc>
        <instances>
        <instance>
        <uri>https://www.googletagservices.com/tag/js/gpt.js</uri>
        <method>GET</method>
        <param></param>
        <attack></attack>
        <evidence>db</evidence>
        <requestheader>GET
https://www.googletagservices.com/tag/js/gpt.js HTTP/1.1 Host:
www.googletagservices.com Connection: keep-alive sec-ch-ua:
&quot;Google Chrome&quot;;v=&quot;107&quot;,
&quot;Chromium&quot;;v=&quot;107&quot;,
&quot;Not=A?Brand&quot;;v=&quot;24&quot; sec-ch-ua-mobile: ?0 User-
Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 sec-ch-ua-platform:
&quot;Windows&quot; Accept: */* Sec-Fetch-Site: cross-site Sec-Fetch-
Mode: no-cors Sec-Fetch-Dest: script Referer: https://jsonlint.com/
Accept-Language: en-US,en;q=0.9 </requestheader>
        <requestbody></requestbody>
        <responseheader>HTTP/1.1 200 OK Vary: Accept-Encoding Content-
Type: text/javascript </responseheader>
        <responsebody>(function(E)) </responsebody>
        </instance>
        <instance>
        <uri>https://www.googletagservices.com/tag/js/gpt.js</uri>
        <method>GET</method>
        <param></param>
        <attack></attack>
        <evidence>query</evidence>
        <requestheader>GET
https://www.googletagservices.com/tag/js/gpt.js HTTP/1.1 Host:
www.googletagservices.com Connection: keep-alive sec-ch-ua:
&quot;Google Chrome&quot;;v=&quot;107&quot;,
&quot;Chromium&quot;;v=&quot;107&quot;,
&quot;Not=A?Brand&quot;;v=&quot;24&quot; sec-ch-ua-mobile: ?0 User-
Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 sec-ch-ua-platform:
&quot;Windows&quot; Accept: */* Sec-Fetch-Site: cross-site Sec-Fetch-
Mode: no-cors Sec-Fetch-Dest: script Referer: https://jsonlint.com/
Accept-Language: en-US,en;q=0.9 </requestheader>
        <requestbody></requestbody>
        <responseheader>HTTP/1.1 200 OK Vary: Accept-Encoding Content-
Type: text/javascript Cross-Origin-Resource-Policy: cross-origin
Cross-Origin-Opener-Policy-Report-Only: same-origin; report-
to=&quot;adsgpt-scs&quot; Report-To: {&quot;group&quot;:&quot;ads-
gptscs&quot;,&quot;max_age&quot;:2592000,&quot;endpoints&quot;:[{&quot
;url&q uot;:&quot;https://csp.withgoogle.com/csp/report-to/ads-gpt-
scs&quot;}]} Timing-Allow-Origin: * Content-Length: 80512 Date: Thu,
17 Nov 2022 05:20:21 GMT Expires: Thu, 17 Nov 2022 05:20:21 GMT Cache-
Control: private, max-age=900, stale-while-revalidate=3600 ETag:
&quot;1394 / 793 of 1000 / last-modified: 1668639967&quot; X-Content-
```

```
Type-Options: nosniff Server: sffe X-XSS-Protection: 0 Alt-Svc:
h3=&quot;:443&quot;; ma=2592000,h3-29=&quot;:443&quot;; ma=2592000,h3-
Q050=&quot;:443&quot;; ma=2592000,h3-Q046=&quot;:443&quot;;
ma=2592000,h3-Q043=&quot;:443&quot;; ma=2592000,quic=&quot;:443&quot;;
ma=2592000; v=&quot;46,43&quot; </responseheader>
        <responsebody>(function(E){var window=this..});
</responsebody>
        </instance>
        </instances>
        <count>2</count>
        <solution>Remove all comments that return information that may
help an attacker and fix any underlying problems they refer
to.</solution>
        <otherinfo>The following pattern was used: \bDB\b and was
detected in the element starting with: &quot;var aa,ba=function(a){var
b=0;return function(){return
b&lt;a.length?{done:!1,value:a[b++]}:{done:!0}}},ca=&quot;function&quo
t;= =typeof Objec&quot;, see evidence field for the suspicious
comment/snippet.</otherinfo>
        <reference></reference>
        <cweid>200</cweid>
        <wascid>13</wascid>
        <sourceid>8</sourceid>
        <tags>
        <tag>
        <tag>OWASP_2021_A01</tag>
        <link>https://owasp.org/Top10/A01_2021Broken_Access_Control/</
link>
        </tag>
        <tag>
        <tag>OWASP_2017_A03</tag>
        <link>https://owasp.org/www-project top-ten/2017/A3_2017-
Sensitive_Data_Exposure.html</link>
        </tag>
        </tags>
        </alertitem>
    </alerts>
    </site>
    </OWASPZAPReport>
    </owaspZapXml>
</data>
</ServiceRequest>
```

XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://
<QualysBaseURL>/qps/xsd/3.0/was/owaspzap.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
    <OwaspZap>
        <id>2001</id>
        <webApp>
        <id>29120395</id>
        <name>
            <![CDATA[Import Zap finding Web app1]]>
        </name>
        <url>
            <![CDATA[http://10.10.60.90]]>
        </url>
        </webApp>
        <alerts>
        <list>
            <AlertItem>
            <alertRef>10027</alertRef>
            </AlertItem>
            <AlertItem>
            <alertRef>10035</alertRef>
            </AlertItem>
        </list>
        </alerts>
        <fileName>testOwaspFile</fileName>
        <errorRecords>
        <count>0</count>
        </errorRecords>
    </OwaspZap>
    </data>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/owaspzap.xsd

Search OWASP ZAP Findings

/qps/rest/3.0/search/was/owaspzapfinding

[POST]

Returns list of OWSP findings found in web applications which are in the user’s scope.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”. WAS Permissions- "Access OWASP ZAP Report" and "Finding read OWASP ZAP ". The output includes findings in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) ID of the finding.

uniqueId	(value) The 36-bit unique id assigned to the finding.
	For example:
	`<Finding>`
	`   <id>132990</id>`
	`   <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>`
	`...`

name	(text) Name of the detection finding.
alertRef	(String)Reference of OWASP ZAP alert.

webApp.id	(Integer) ID of the web application on which the finding was detected.
webApp.name	(String) Name of the web application on which the finding was detected.
webApp.tags	(Integer) The tags associated with the web application being scanned. Note: This parameter supports operator=″NONE″.
webApp.tags.id	(Integer) The tag ID assigned to web application being scanned.
webApp.tags.name	(String) Name of the tag associated with the web application on which the finding was detected.

Sample - Search for finding with specific ID

API request

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

Request POST data

```
<ServiceRequest>
   <filters>
     <Criteria field="id" operator="EQUALS">1002</Criteria>
   </filters>
</ServiceRequest>
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://QualysBaseURL>
/qps/xsd/3.0/was/owaspzapfinding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <OwaspZapFinding>
            <id>1002</id>
            <uniqueId>f2b03430-87d5-450b-98c5-6ce210b41e8c</uniqueId>
            <findingType>OWASPZAP</findingType>
            <pluginid>10035</pluginid>
            <alertRef>10035</alertRef>
            <alert>Strict-Transport-Security Header Not Set</alert>
            <name>Strict-Transport-Security Header Not Set</name>
            <riskcode>1</riskcode>
            <confidence>3</confidence>
            <riskdesc>Low (High)</riskdesc>
            <confidencedesc>High</confidencedesc>
            <desc>
HTTP Strict Transport Security (HSTS) is a web security policy
mechanism whereby a web server declares that complying user agents
(such as a web browser) are to interact with it using only secure
HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF
standards track protocol and is specified in RFC 6797.
</desc>
            <count>1</count>
            <solution>
Ensure that your web server, application server, load balancer, etc.
is configured to enforce Strict-TransportSecurity.</solution>
            <reference>https://cheatsheetseries.owasp.org/cheatsheets/
HTTP_Strict_Tra nsport_Security_Cheat_Sheet.html
https://owasp.org/www-community/Security_Headers
http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
http://caniuse.com/stricttransportsecurity
http://tools.ietf.org/html/rfc6797</reference>
            <cweid>319</cweid>
            <wascid>15</wascid>
            <sourceid>8</sourceid>
            <tags>
                <list>
                    <OwaspZapTag>
                        <tag>OWASP_2021_A05</tag>
                        <link>https://owasp.org/Top10/A05_2021Security
_Misconfiguration/</link>
                    </OwaspZapTag>
                    <OwaspZapTag>
                        <tag>OWASP_2017_A06</tag>
                        <link>https://owasp.org/www-project-
```

```
topten/2017/A6_2017-Security_Misconfiguration.html</link>
                    </OwaspZapTag>
                </list>
            </tags>
            <instances>
                <list>
                    <Instance>
                        <uri>https://www.googletagservices.com/tag/js/
gpt.js</uri>
                        <method>GET</method>
                        <requestheader>GET
https://www.googletagservices.com/tag/js/gpt.js HTTP/1.1 Host:
www.googletagservices.com Connection: keep-alive sec-ch-ua:
&quot;Google Chrome&quot;;v=&quot;107&quot;,
&quot;Chromium&quot;;v=&quot;107&quot;,
&quot;Not=A?Brand&quot;;v=&quot;24&quot; sec-ch-ua-mobile: ?0 User-
Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 sec-ch-ua-platform:
&quot;Windows&quot; Accept: */* Sec-Fetch-Site: cross-site Sec-Fetch-
Mode: no-cors Sec-Fetch-Dest: script Referer: https://jsonlint.com/
Accept-Language: en-US,en;q=0.9
</requestheader>
                        <responseheader>HTTP/1.1 200 OK Vary: Accept-
Encoding Content-Type: text/javascript Cross-Origin-Resource-Policy:
cross-origin Cross-Origin-Opener-Policy-Report-Only: same-origin;
report-to=&quot;adsgpt-scs&quot; Report-To:
{&quot;group&quot;:&quot;ads-
gptscs&quot;,&quot;max_age&quot;:2592000,&quot;endpoints&quot;:[{&quot
;url&q uot;:&quot;https://csp.withgoogle.com/csp/report-to/ads-gpt-
scs&quot;}]}Qualys Cloud Platform v3.x WAS API: New API for OWASP ZAP
Findings 61 Timing-Allow-Origin: * Content-Length: 80512 Date: Thu, 17
Nov 2022 05:20:21 GMT Expires: Thu, 17 Nov 2022 05:20:21 GMT Cache-
Control: private, max-age=900, stale-while-revalidate=3600 ETag:
&quot;1394 / 793 of 1000 / last-modified: 1668639967&quot; X-Content-
Type-Options: nosniff Server: sffe X-XSS-Protection: 0 Alt-Svc:
h3=&quot;:443&quot;; ma=2592000,h3-29=&quot;:443&quot;; ma=2592000,h3-
Q050=&quot;:443&quot;; ma=2592000,h3-Q046=&quot;:443&quot;;
ma=2592000,h3-Q043=&quot;:443&quot;; ma=2592000,quic=&quot;:443&quot;;
ma=2592000; v=&quot;46,43&quot;
</responseheader>
                    </Instance>
                </list>
            </instances>
        </OwaspZapFinding>
```

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

XSD

<platform API server>/qps/xsd/3.0/was/owaspzap.xsd

Finding OWASP ZAP Count

/qps/rest/3.0/count/was/owaspzapfinding

[POST]

Returns the count of OWASP ZAP findings for a selected criteria.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”. WAS Permissions "Access OWASP ZAP Report". The count includes web applications in the user's scope.

Input Parameters

These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND.

Click here for available operators

Parameter	Description
id	(integer) ID of the finding.

uniqueId	(value) The 36-bit unique id assigned to the finding.
	For example:
	`<Finding>`
	`   <id>132990</id>`
	`   <uniqueId>8a2c4d51-6d28-2b92-e053-2943720a74ab</uniqueId>`
	`...`

name	(text) Name of the detection finding.
alertRef	(String)Reference of OWASP ZAP alert.
webApp.id	(Integer) ID of the web application on which the finding was detected.

webApp.name	(String) Name of the web application on which the finding was detected.
webApp.tags	(Integer) The tags associated with the web application being scanned. Note: This parameter supports operator="NONE".
webApp.tags.id	(Integer) The tag ID assigned to web application being scanned.
webApp.tags.name	(String) Name of the tag associated with the web application on which the finding was detected.

Sample - Get count of findings with a criteria

API request

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

Request POST data

```
<ServiceRequest>
    <filters>
        <Criteria field="id" operator="EQUALS">1002</Criteria>
    </filters>
</ServiceRequest>
```

XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"xsi:noNamespaceSchemaLocation="https://
    <QualysBaseURL>/qps/xsd/3.0/was/ow aspzapfinding.xsd">
       <responseCode>SUCCESS</responseCode>
        <count>1</count>
</ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/owaspzapfinding.xsd

Get OWASP ZAP Finding Details

/qps/rest/3.0/get/was/owaspzapfinding/<id>/

[GET]

Returns details of a specific OWASP ZAP finding.

Permissions required - User must have WAS module enabled. User account must have these permissions: Access Permission “API Access”. WAS Permissions- "Access OWASP ZAP Report" and "Finding read OWASP ZAP". The output includes findings for web applications in the user's scope.

Input Parameters

The element “id” (integer) is required, where “id” identifies finding id of OWASP ZAP finding.

Click here for available operators

Sample - View details for the finding

Let us view details for the OWASP ZAP finding with the ID 1001.

API request

```
curl -n -u "USERNAME:PASSWORD"
"https:///qps/rest/3.0/get/was/owaspzapfinding/1001"
```

XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://
    <QualysBaseURL>/qps/xsd/3.0/was/ow aspzapfinding.xsd">
        <responseCode>SUCCESS</responseCode>
        <count>1</count>
        <data>
            <OwaspZapFinding>
                <id>1001</id>
                <uniqueId>a2e825d4-db9d-49a9-842a-
```

```
4c22fab555eb</uniqueId>
                <findingType>OWASPZAP</findingType>
                <pluginid>10027</pluginid>
                <alertRef>10027</alertRef>
                <alert>Information Disclosure - Suspicious
Comments</alert>
                <name>Information Disclosure - Suspicious
Comments</name>
                <riskcode>0</riskcode>
                <confidence>1</confidence>
                <riskdesc>Informational (Low)</riskdesc>
                <confidencedesc>Low</confidencedesc>
                <desc>The response appears to contain suspicious
comments which may help an attacker. Note: Matches made within script
blocks or files are against the entire content not only
comments.</desc> API affectedqps/rest/3.0/get/was/owaspzapfinding/
                <id> New or Updated APIsNew OperatorGET DTD or XSD
changesYes
                    <count>2</count>
                    <solution>Remove all comments that return
information that may help an attacker and fix any underlying problems
they refer to.</solution>
                    <otherinfo>The following pattern was used: \bDB\b
and was detected in the element starting with: &quot;var
aa,ba=function(a){var b=0;return function(){return
b&amp;lt;a.length?{done:!1,value:a[b++]}:{done:!0}}},ca=&quot;function
&qu ot;==typeof Objec&quot;, see evidence field for the suspicious
comment/snippet.</otherinfo>
                    <cweid>200</cweid>
                    <wascid>13</wascid>
                    <sourceid>8</sourceid>
                    <tags>
                        <list>
                            <OwaspZapTag>
                                <tag>OWASP_2017_A03</tag>
                                <link>https://owasp.org/www-project-
topten/2017/A3_2017-Sensitive_Data_Exposure.html</link>
                            </OwaspZapTag>
                            <OwaspZapTag>
                                <tag>OWASP_2021_A01</tag>
                                <link>https://owasp.org/Top10/A01_2021
Broken_Access_Control/</link>
                            </OwaspZapTag>
                        </list>
                    </tags>
                    <instances>
                        <list>
                            <Instance>
                                <uri>https://www.googletagservices.com
/tag/js/gpt.js</uri>
                                <method>GET</method>
                                <evidence>query</evidence>
                                <requestheader>GET
https://www.googletagservices.com/tag/js/gpt.js HTTP/1.1 Host:
www.googletagservices.com Connection: keep-alive sec-ch-ua:
&quot;Google Chrome&quot;;v=&quot;107&quot;,
&quot;Chromium&quot;;v=&quot;107&quot;,
&quot;Not=A?Brand&quot;;v=&quot;24&quot; sec-ch-ua-mobile: ?0 User-
Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 sec-ch-ua-platform:
&quot;Windows&quot; Accept: */* Sec-Fetch-Site: cross-site Sec-Fetch-
Mode: no-cors Sec-Fetch-Dest: script Referer: https://jsonlint.com/
Accept-Language: en-US,en;q=0.9 </requestheader>
                                <responseheader>HTTP/1.1 200 OK Vary:
Accept-Encoding Content-Type: text/javascript Cross-Origin-Resource-
Policy: cross-origin Cross-Origin-Opener-Policy-Report-Only: same-
origin; report-to=&quot;adsgpt-scs&quot; Report-To:
{&quot;group&quot;:&quot;ads-
gptscs&quot;,&quot;max_age&quot;:2592000,&quot;endpoints&quot;:[{&quot
;url&q uot;:&quot;https://csp.withgoogle.com/csp/report-to/ads-gpt-
scs&quot;}]} Timing-Allow-Origin: * Content-Length: 80512 Date: Thu,
17 Nov 2022 05:20:21 GMT Expires: Thu, 17 Nov 2022 05:20:21 GMT Cache-
Control: private, max-age=900, stale-while-revalidate=3600 ETag:
&quot;1394 / 793 of 1000 / last-modified: 1668639967&quot; X-Content-
Type-Options: nosniff Server: sffe X-XSS-Protection: 0 Alt-Svc:
h3=&quot;:443&quot;; ma=2592000,h3-29=&quot;:443&quot;; ma=2592000,h3-
Q050=&quot;:443&quot;; ma=2592000,h3-Q046=&quot;:443&quot;;
ma=2592000,h3-Q043=&quot;:443&quot;; ma=2592000,quic=&quot;:443&quot;;
ma=2592000; v=&quot;46,43&quot; </responseheader>
                            </Instance>
                            <Instance>
                                <uri>https://www.googletagservices.com
/tag/js/gpt.js</uri>
                                <method>GET</method>
                                <evidence>db</evidence>
                                <requestheader>GET
https://www.googletagservices.com/tag/js/gpt.js HTTP/1.1 Host:
www.googletagservices.com Connection: keep-alive sec-ch-ua:
&quot;Google Chrome&quot;;v=&quot;107&quot;,
&quot;Chromium&quot;;v=&quot;107&quot;,
&quot;Not=A?Brand&quot;;v=&quot;24&quot; sec-ch-ua-mobile: ?0 User-
Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 sec-ch-ua-platform:
&quot;Windows&quot; Accept: */* Sec-Fetch-Site: cross-site Sec-Fetch-
Mode: no-cors Sec-Fetch-Dest: script Referer: https://jsonlint.com/
Accept-Language: en-US,en;q=0.9 </requestheader>
                                <responseheader>HTTP/1.1 200 OK Vary:
Accept-Encoding Content-Type: text/javascript Cross-Origin-Resource-
Policy: cross-origin Cross-Origin-Opener-Policy-Report-Only: same-
origin; report-to=&quot;adsgpt-scs&quot; Report-To:
{&quot;group&quot;:&quot;ads-
gptscs&quot;,&quot;max_age&quot;:2592000,&quot;endpoints&quot;:[{&quot
;url&q uot;:&quot;https://csp.withgoogle.com/csp/report-to/ads-gpt-
scs&quot;}]} Timing-Allow-Origin: * Content-Length: 80512 Date: Thu,
17 Nov 2022 05:20:21 GMT Expires: Thu, 17 Nov 2022 05:20:21 GMT Cache-
Control: private, max-age=900, stale-while-revalidate=3600 ETag:
&quot;1394 / 793 of 1000 / last-modified: 1668639967&quot; X-Content-
Type-Options: nosniff Server: sffe X-XSS-Protection: 0 Alt-Svc:
h3=&quot;:443&quot;; ma=2592000,h3-29=&quot;:443&quot;; ma=2592000,h3-
Q050=&quot;:443&quot;; ma=2592000,h3-Q046=&quot;:443&quot;;
ma=2592000,h3-Q043=&quot;:443&quot;; ma=2592000,quic=&quot;:443&quot;;
ma=2592000; v=&quot;46,43&quot; </responseheader>
                            </Instance>
                        </list>
                    </instances>
                </OwaspZapFinding>
            </data>
        </ServiceResponse>
```

XSD

<platform API server>/qps/xsd/3.0/was/owaspzapfinding.xsd

Error Messages

Sample Messages: Elements

Sample messages for element errors are shown below

Error Message	Resolution
Element Validation	
url: Invalid URL format (<value>).	URL format must be as follows: http://<baseUrl>/rest/3.0/?parameters
<scope>: Invalid value (<value>).	Element must be set to one of these values: ALL, LIMIT, SUBDOMAIN or DOMAINS.
domains: Element is required when scope is set to: DOMAINS.	Specify the domains to include in the web application scope in the “domains” element.
subDomain: Element is required when scope is set to: SUBDOMAIN.	Specify the subdomains to include in the web application scope in the “subDomain” element.
subDomain: Invalid domain name format (<value>).	Use following format in the “subDomain” element: .my.domain.suffix (must start with a dot)
useRobots: Invalid value (<value>).	Element “userRobots” must be set to one of these values: IGNORE, ADD_PATHS, EXCLUDELISt.
Url: Element is required	Element “Url” is required.

uris.<field>: Invalid URL format (<value>).	For the uri.<field> sub element, specify a URL like http://domain.name/base/url/?parameters
uris.<field>: Length of the field must not be greater than 2048 characters. (<value>).	For the uri.<field> sub element, the maximum field length is 2048 characters.
Domain: Element is required	The domain element must be provided.
Domain: Invalid host name format (<value>).	Use following format for value in the "Domain" element: www.my.domain.example.
Length of all domains cannot exceed 2048 characters.	The list of all domains in the web application cannot exceed 2048 characters.
Attribute.category: Element is required.	The element Attribute.category is required.
Attribute.category: Invalid value (<value>).	Element Attribute.category must be set to one of these values: Business Function, Business Location, Business Description.
Attribute.value: Element is required.	Provide a value for the attribute in the Attribute.value element: function, location or description.
The attribute length cannot be greater than 64 characters.	The value for this attribute cannot exceed 64 characters.

The attribute length cannot be greater than 2048 characters.	The value for this attribute cannot exceed 2048 characters.
<element>: Element must not be set.	This element does not apply to this request.
set: Element must contain at least one child.	The set element requires at least one sub element.
At least one of the following elements must be set: set, add, remove.	This request requires at least one of these elements: set, add or remove.
headers: Length of all headers cannot exceed 2048 characters.	The values of all headers cannot exceed 2048 characters.
At least one of the following elements must be set: set, add, remove.	For an “update” request you must set at least one of these elements: set, add or remove.
UrlEntry: Element is required.	The element UrlEntry must be provided.
UrlEntry: Invalid URL format (value).	Specify a URL like http://domain.name/base/url/?parameters
<parent>: Length of all [URLs, regular expressions] cannot exceed 2048 characters	The list of entries for a given type shall not exceed 2048 characters.

UrlEntry: Only regular expressions are accepted for this element.	You must provide regular expressions for the element postDataExcludelist.
tags.<element>: Element must not be set.	The tags element does not apply for this request
tags.set: Element must contain at least one child.	At least one sub element must be provided for the element tag.set.
Tag.id: Element is required.	Provide a value for the element Tag.id
Tag.id: Invalid value (value).	Value must be an integer set at least to 1.
Tag: Tag specified by ID <id> does not exist or is not available.	Provide a value for the element id that corresponds to a valid tag.

Sample Messages: Authorization

Sample messages for errors related to authorization are shown below.

Error Message	Resolution
Element Validation	
You are not authorized to access the application through the API.	You must be granted the API Access permission in your roles and scopes.
You do not have access to module Web Application Scanning required by this API.	Please contact your account manager to have WAS enabled in your subscription.
No data shall be passed for this operation.	The POST request does not specify a data element.
User is not authorized to perform this operation on specified object(s).	You must be granted access to these objects in your user scope.
Operation %s does not support search filters.	Do not provide search filers for this operation
Quota of web application has been exceeded.	Please check with your account manager to purchase new applications.

Sample Messages: Criteria

Sample messages for errors related to criteria are shown below.

Error Message	Resolution
Element Validation	
Criteria: Field is required.	Specify the name of the criteria to search against.
Criteria: Invalid criteria (<field name>).	Please search against one of the following criteria: %s.
Criteria: Invalid operator for criteria '<field>' (<operator>).	Allowed operations for this criteria are: %s.
Criteria: Value is required for criteria '<field>'.	Specify a value for a field name for search criteria.
Criteria: Invalid value format for criteria '<field>': <value>.	Boolean (true, false). Date and Time in UTC format Enumeration (allowed options separated by comma). Other: Specify criteria value(s) as <type>.

Sample Messages: Report Storage Limit

Sample messages for errors related to report storage limit are shown below.

Error Message	Resolution
Element Validation	
Your [subscription|user] storage limit of <NB> Mb has been reached.	Delete existing reports and try again.

Available operators

Operators supported by input parameters:

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

Text - CONTAINS, EQUALS, NOT EQUALS

Date - EQUALS, NOT EQUALS, GREATER, LESSER

Keyword - EQUALS, NOT EQUALS, IN

Boolean (true/false) - EQUALS, NOT EQUALS