# Table of Contents

Get Started .................................................................................................................................... 7

Web Application Scanning API ................................................................................................. 7

Qualys user account.................................................................................................................... 8

Making API Calls......................................................................................................................... 9

URL to Qualys API server.......................................................................................................... 12

Tracking API usage by user......................................................................................................... 14

How to Download Vulnerability Details.................................................................................. 15

Know your portal version.......................................................................................................... 20

JSON Support.............................................................................................................................. 22

Web Applications.......................................................................................................................... 26

Count web applications............................................................................................................. 26

Search Web Applications........................................................................................................... 29

Get Web Application Details..................................................................................................... 34

Create Web Application............................................................................................................. 47

Update Web Application............................................................................................................. 72

Delete Web Application............................................................................................................. 80

Purge Web Application.............................................................................................................. 84

Download Selenium Script......................................................................................................... 88

Reference: WebApp..................................................................................................................... 91

Authentication............................................................................................................................... 99

Authentication Count.................................................................................................................. 99

Search Authentication Record.................................................................................................... 102
Get Authentication Record Details ................................................................. 106
Create Authentication Record ...................................................................... 115
Update Authentication Record .................................................................... 129
Delete Authentication Record .................................................................... 131
Reference: Authentication .......................................................................... 135

Scans ........................................................................................................... 137

Scan Count ................................................................................................. 137
Search Scans .............................................................................................. 142
Get Scan Details ......................................................................................... 153
Launch Scans (Single) ................................................................................ 160
Launch Scan (Multiple) ............................................................................... 171
Retrieve Scan Status .................................................................................. 179
Retrieve Scan Results ................................................................................ 181
Cancel Scan ............................................................................................... 187
Delete Scan ............................................................................................... 189

WasScan Reference .................................................................................... 192
WAS Scan Results Reference ...................................................................... 198

Schedules .................................................................................................. 200

Schedule Count ......................................................................................... 200
Search Schedule ....................................................................................... 205
Get Schedule Details .................................................................................. 214
Create a Schedule (single web application) ............................................ 218
Create Schedules (Multiple) ..................................................................... 232
Update Schedule ............................................................................................................... 246
Activate an Existing Schedule ...................................................................................... 249
Deactivate Schedule ......................................................................................................... 253
Delete Schedule .................................................................................................................. 257
Download Schedule ......................................................................................................... 260
Reference: Schedule ........................................................................................................ 263
Reports ...................................................................................................................................... 265
Report Count ...................................................................................................................... 265
Search Report ..................................................................................................................... 268
Get Report Details ............................................................................................................. 273
Get Report Status .............................................................................................................. 275
Download Report ............................................................................................................... 277
Send Encrypted PDF Report ......................................................................................... 278
Update Report .................................................................................................................... 280
Delete Report ....................................................................................................................... 282
Report Creation ...................................................................................................................... 285
Create Report ...................................................................................................................... 285
Web Application Report ................................................................................................ 286
Scan Report ........................................................................................................................ 293
Scorecard Report .............................................................................................................. 303
Catalog Report .................................................................................................................... 310
Report Template Count .................................................................................................... 315
Search Report Template .................................................................................................. 317
Get details of Report Template .......................................................................................... 321
Reference: Report ............................................................................................................. 324

Option Profiles .................................................................................................................. 327
Option Profile Count ......................................................................................................... 327
Search Option Profiles .................................................................................................... 330
Get Option Profile Details ............................................................................................... 333
Create a new Option Profile ........................................................................................... 341
Update an Option Profile ................................................................................................ 361
Delete an Option Profile .................................................................................................. 372
Reference: Option Profile ............................................................................................... 376

Findings ............................................................................................................................. 379
Finding Count ..................................................................................................................... 379
Search Findings ................................................................................................................ 383
Get Finding Details ........................................................................................................... 390
Ignore Findings .................................................................................................................. 394
Activate Findings ............................................................................................................. 400
Edit Finding Severity ....................................................................................................... 404
Restore Findings Severity .............................................................................................. 407
Retest Findings ................................................................................................................... 410
WAS Findings in XML Report ........................................................................................ 412
Reference: Findings ........................................................................................................... 418

Burp ..................................................................................................................................... 421
Import Burp Scan Report ................................................................................................. 421
Error Messages ....................................................................................................................... 427

Sample Messages: Elements ......................................................................................... 427

Sample Messages: Authorization ................................................................................. 431

Sample Messages: Criteria ............................................................................................. 432

Sample Messages: Report Storage Limit ................................................................. 433
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.

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

```xml
<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

```xml
<ServiceRequest>
  <filters> ...
  </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

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

<table>
<thead>
<tr>
<th>Account location</th>
<th>API server URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualys US Platform 1</td>
<td><a href="https://qualysapi.qualys.com">https://qualysapi.qualys.com</a></td>
</tr>
<tr>
<td>Qualys US Platform 2</td>
<td><a href="https://qualysapi.qg2.apps.qualys.com">https://qualysapi.qg2.apps.qualys.com</a></td>
</tr>
<tr>
<td>Qualys US Platform 3</td>
<td><a href="https://qualysapi.qg3.apps.qualys.com">https://qualysapi.qg3.apps.qualys.com</a></td>
</tr>
<tr>
<td>Qualys EU Platform 1</td>
<td><a href="https://qualysapi.qualys.eu">https://qualysapi.qualys.eu</a></td>
</tr>
<tr>
<td>Qualys EU Platform 2</td>
<td><a href="https://qualysapi.qg2.apps.qualys.eu">https://qualysapi.qg2.apps.qualys.eu</a></td>
</tr>
<tr>
<td>Qualys India Platform 1</td>
<td><a href="https://qualysapi.qg1.apps.qualys.in">https://qualysapi.qg1.apps.qualys.in</a></td>
</tr>
<tr>
<td>Qualys Private Cloud Platform</td>
<td><a href="https://qualysapi">https://qualysapi</a>.&lt;customer_base_url&gt;</td>
</tr>
</tbody>
</table>

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.sjdbc01.qualys.com:443`
  - `nchost.sjdbc01.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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action=list</td>
<td>(Required) A flag used to request the download of vulnerability data from the KnowledgeBase.</td>
</tr>
<tr>
<td>echo_request={0|1}</td>
<td>(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.</td>
</tr>
<tr>
<td>details={Basic|All|None}</td>
<td>(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.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ids={value}</td>
<td>(Optional) Used to filter the XML output to include only vulnerabilities that have QID numbers matching the QID numbers you specify.</td>
</tr>
<tr>
<td>id_min={value}</td>
<td>(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.</td>
</tr>
<tr>
<td>id_max={value}</td>
<td>(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.</td>
</tr>
<tr>
<td>is_patchable={0</td>
<td>1}</td>
</tr>
<tr>
<td>last_modified_after={date}</td>
<td>(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).</td>
</tr>
<tr>
<td>last_modified_before={date}</td>
<td>(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-</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>last_modified_by_</td>
<td>(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).</td>
</tr>
<tr>
<td>user_after={date}</td>
<td></td>
</tr>
<tr>
<td>last_modified_by_</td>
<td>(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).</td>
</tr>
<tr>
<td>user_before={date}</td>
<td></td>
</tr>
<tr>
<td>last_modified_by_</td>
<td>(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).</td>
</tr>
<tr>
<td>service_after={date}</td>
<td></td>
</tr>
<tr>
<td>last_modified_by_</td>
<td>(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).</td>
</tr>
<tr>
<td>service_before={date}</td>
<td></td>
</tr>
<tr>
<td>published_after={date}</td>
<td>(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).</td>
</tr>
<tr>
<td>published_before={date}</td>
<td>(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).</td>
</tr>
<tr>
<td>discovery_method={value}</td>
<td>(Optional) Used to filter the XML output to show only vulnerabilities assigned a certain discovery method. A valid value is: Remote, Authenticated, RemoteOnly,</td>
</tr>
</tbody>
</table>
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 KnowledgeBase, all details

API request

```bash
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.com/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

```bash
```

Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Portal-Version>
      <WAS-VERSION>6.0.0.0</WAS-VERSION>
      <FIM-VERSION>1.5.1</FIM-VERSION>
      <VM-VERSION>1.0.3</VM-VERSION>
      <CERTVIEW-VERSION>1.1.0.0</CERTVIEW-VERSION>
      <CM-VERSION>1.20.1</CM-VERSION>
      <MDS-VERSION>2.11.7.0</MDS-VERSION>
      <CA-VERSION>2.9.1.0</CA-VERSION>
      <IOC-VERSION>1.1.0</IOC-VERSION>
      <AV2-VERSION>0.1.0</AV2-VERSION>
      <QUESTIONNAIRE-VERSION>2.14.0.4</QUESTIONNAIRE-VERSION>
      <WAF-VERSION>2.7.0.0</WAF-VERSION>
    </Portal-Version>
  </data>
</ServiceResponse>
```
Sample JSON

API request

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

Response

```
{
    "ServiceResponse": {
        "data": [
            {
                "Portal-Version": {
                    "WAS-VERSION": "6.0.0.0",
                    "VM-VERSION": "1.0.3",
                    "CM-VERSION": "1.20.1",
                    "MDS-VERSION": "2.11.7.0",
                    "CA-VERSION": "2.9.1.0",
                    "QUESTIONNAIRE-VERSION": "2.14.0.4",
                    "WAF-VERSION": "2.7.0.0"
                },
                ...
            }
        ],
        "responseCode": "SUCCESS",
        "count": 1
    }
}
```
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**

```
"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
          }
        }
      }
    ]
  }
}
```
{"count": 1,
"responseCode": "SUCCESS",

"OP creation - with json request and response",
"ignoreBinaryFiles": false,
"timeoutErrorThreshold": 10,
"performance": "LOW",
"name": "OP creation - with json request and response",
"ignoreBinaryFiles": "false",
"unexpectedErrorThreshold": 20,
"isDefault": false,
"parameterSet": {
"name": "Initial Parameters",
"id": 0
},
"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
}
Sample 2 - Launch a scan

API request


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
"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
   }
}
Web Applications

Count web applications

/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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Web application ID.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Web application name.</td>
</tr>
<tr>
<td>url</td>
<td>(text) The URL of web application.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Tag name assigned to web application.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) Tag ID assigned to web application.</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the web application was created</td>
</tr>
</tbody>
</table>
**Qualys Web Application Scanning API**

**Web Applications**

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**

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

**XML response**

```xml
<?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 Applications

/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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Web application ID.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Web application name.</td>
</tr>
<tr>
<td>url</td>
<td>(text) The URL of web application.</td>
</tr>
<tr>
<td>tags</td>
<td>(element) Tags assigned to web application. Click here for description of this &lt;WebApp&gt; element</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Tag name assigned to web application.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) Tag ID assigned to web application.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the web application was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>updatedDate</td>
<td>(date) The date when the web application was last updated in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>isScheduled</td>
<td>(boolean) A flag indicating whether a scan is scheduled for web application.</td>
</tr>
<tr>
<td>isScanned</td>
<td>(boolean) A flag indicating whether the web application has been scanned.</td>
</tr>
<tr>
<td>lastScan.status</td>
<td>(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED</td>
</tr>
<tr>
<td>lastScan.date</td>
<td>(date) Date when web application was last scanned, in UTC date/time format.</td>
</tr>
</tbody>
</table>

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/webapp" -X "POST"

**XML response**

```xml
<?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>
    </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>
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).
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 (.png). In order to encode screenshots we use urlSafe Base 64 encoding solution like other elements in our API. Therefore these characters will be replaced in the base64 contents:

/ will be replaced with _

+ will be replaced with -

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
Qualys Web Application Scanning API
Web Applications

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.acme01.acme.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>acme_as</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
      <scope>ALL</scope>
      <attributes>
        <count>0</count>
      </attributes>
    </WebApp>
  </data>
</ServiceResponse>
```
<id>139359</id>
<name><![CDATA[10 Links edit]]></name>
</defaultProfile>
<defaultScanner>
  <type>EXTERNAL</type>
</defaultScanner>
<scannerLocked>false</scannerLocked>
?urlBlacklist>
  <count>0</count>
</urlBlacklist>
?urlWhitelist>
  <count>0</count>
</urlWhitelist>
<postDataBlacklist>
  <count>0</count>
</postDataBlacklist>
<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>
    </monthlyOccurrence>
  </occurrence>
</malwareScheduling>
<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>acme_as</username>
  <firstName><![CDATA[Alex]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</createdBy>
<createdDate>2017-07-24T09:08:49Z</createdDate>
(updatedBy>
  <id>4354</id>
  <username>acme_as</username>
  <firstName><![CDATA[Alex]]></firstName>
</updatedBy>
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/cassium/xss/]]>
      </url>
      <owner>
        <id>4354</id>
        <username>acme_ab2</username>
      </owner>
    </WebApp>
  </data>
</ServiceResponse>
<firstName>Adam</firstName>
<lastName>Smith</lastName>
<scope>ALL</scope>
<attributes>
  <count>0</count>
</attributes>
<defaultScanner>
  <type>INTERNAL</type>
  <friendlyName>db4_acme_ab2</friendlyName>
</defaultScanner>
<scannerLocked>false</scannerLocked>
<progressiveScanning>true</progressiveScanning>
=urlBlacklist
  <count>0</count>
</urlBlacklist>
=urlWhitelist
  <count>0</count>
</urlWhitelist>
=postDataBlacklist
  <count>0</count>
</postDataBlacklist>
<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>
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>
      <urlBlacklist>
        <count>0</count>
      </urlBlacklist>
    </WebApp>
  </data>
</ServiceResponse>
Sample - Default authentication record details

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

API request

```bash
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
<?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>
      ...
    </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
<?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/xsd/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>
    </WebApp>
  </data>
</ServiceResponse>
```
</defaultScanner>
<scannerLocked>false</scannerLocked>
?urlBlacklist>
  <count>0</count>
</urlBlacklist>
?urlWhitelist>
  <count>0</count>
</urlWhitelist>
postDataBlacklist>
  <count>0</count>
</postDataBlacklist>
<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>
</crawlingScripts>
<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 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>
    </tbody>
</body>
</html>]]></data>
</SeleniumScript>
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

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

XML response

```xml
<ServiceResponse>

```
Qualys Web Application Scanning API

Web Applications

```xml
<responseCode>SUCCESS</responseCode>
<count>1</count>
<data>
  <WebApp>
    <id>323102</id>
    <name><![CDATA[MamboCMS]]></name>
    <url><![CDATA[http://funkytown.acme01.acme.com/Forms/FormFields/temp/updated_web_app_name]]></url>
  </WebApp>
  ...
  <scannerLocked>false</scannerLocked>
  <progressiveScanning>false</progressiveScanning>
  ...
</data>
```

XSD

```xml
<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.

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 blacklists or whitelists 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

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

```bash
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

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

```xml
<?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>
      <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>
      <urlBlacklist>
        <count>0</count>
      </urlBlacklist>
      <urlWhitelist>
        <count>0</count>
      </urlWhitelist>
      <postDataBlacklist>
        <count>0</count>
      </postDataBlacklist>
      <authRecords>
        <count>0</count>
      </authRecords>
      <useRobots>IGNORE</useRobots>
      <useSitemap>false</useSitemap>
      <malwareMonitoring>false</malwareMonitoring>
      <tags>
        <count>0</count>
      </tags>
      <comments>
```
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>
      </authRecords>
    </WebApp>
  </data>
</ServiceRequest>
XML response

```xml
<?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>
            <urlBlacklist>
                <count>0</count>
            </urlBlacklist>
            <urlWhitelist>
                <count>0</count>
            </urlWhitelist>
            <postDataBlacklist>
                <count>0</count>
            </postDataBlacklist>
            <authRecords>
                <count>1</count>
            </authRecords>
        </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>
        </set>
        <domains>
        <set>
          <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>
          </set>
        </uris>
  </data>
</ServiceResponse>
```
Qualys Web Application Scanning API

Web Applications

<Url><![CDATA[https://corp1.myapp.com/]]></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?param=true]]></Url>
<Url><![CDATA[http://corp1.myapp.com/startingUri?param=true&param2=false]]></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:443]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com:8080]]></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/startingUri?param=true&param2=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>
</defaultProfile>
</defaultScanner>
<defaultScanner>
<type>INTERNAL</type>
</defaultScanner>
<friendlyName><![CDATA[friendlyname]]></friendlyName>
</defaultScanner>
<dnsOverrides>
</dnsOverrides>
<set>
  <DnsOverride>
    <id>2022</id>
  </DnsOverride>
</set>
</dnsOverrides>
<useRobots>BLACKLIST</useRobots>
<useSitemap>true</useSitemap>
<headers>
  <set>
    <WebAppHeader><![CDATA[some headers]]></WebAppHeader>
  </set>
</headers>
<urlBlacklist>
  <set>
    <UrlEntry regex="true">
      <![CDATA[http://rg.blacklist.*.qa.myapp.com]]>
    </UrlEntry>
    <UrlEntry regex="true">
      <![CDATA[http://rg.blacklist.*?]]>
    </UrlEntry>
    <UrlEntry>
      <![CDATA[http://url.blacklist.2.ab.myapp.com]]>
    </UrlEntry>
    <UrlEntry regex="false">
      <![CDATA[http://url.blacklist.3.qa.myapp.com]]>
    </UrlEntry>
  </set>
</urlBlacklist>
<urlWhitelist>
  <set>
    <UrlEntry regex="true">
      <![CDATA[http://rg.whitelist.*.qa.myapp.com]]>
    </UrlEntry>
    <UrlEntry regex="true">
      <![CDATA[http://rg.whitelist.*?]]>
    </UrlEntry>
    <UrlEntry>
      <![CDATA[http://url.whitelist.2.ab.myapp.com]]>
    </UrlEntry>
    <UrlEntry regex="false">
      <![CDATA[http://url.whitelist.3.ab.myapp.com]]>
    </UrlEntry>
  </set>
</urlWhitelist>
postDataBlacklist>
  <set>
    <UrlEntry regex="true">
      <![CDATA[http://rg.postdatblacklist.*.ab.myapp.com]]>
    </UrlEntry>
    <UrlEntry regex="true">
      <![CDATA[http://rg.postdatblacklist.*?]]>
    </UrlEntry>
  </set>
</postDataBlacklist>
<comments>
<set>
  <Comment>
    <contents><![CDATA[some additional comments]]></contents>
  </Comment>
</set>
</comments>
</WebApp>
</data>
</ServiceRequest>

**XML response**

```xml
<?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]]></Url>
          <Url><![CDATA[http://corp1.myapp.com]]></Url>
        </list>
      </uris>
    </WebApp>
  </data>
</ServiceResponse>
```
Qualys Web Application Scanning API

Web Applications

<Url><![CDATA[https://corp1.myapp.com]]></Url>
<Url><![CDATA[http://corp1.myapp.com/startingUri?param=1]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com/otherUri?param=1]]></Url>
<Url><![CDATA[https://corp1.myapp.com:443]]></Url>
<Url><![CDATA[http://corp1.myapp.com/startingUri?param=true&param2=false]]></Url>
<Url><![CDATA[http://corp1.myapp.com/otherUri]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com:443]]></Url>
<Url><![CDATA[https://corp2.ab.myapp.com:8080/otherUri]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com:8080/startingUri?param=true&param2=false]]></Url>
<Url><![CDATA[https://corp1.myapp.com/startingUri]]></Url>
<Url><![CDATA[http://corp1.myapp.com/otherUri?param=true]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com/startingUri]]></Url>
<Url><![CDATA[http://corp1.myapp.com]]></Url>
<Url><![CDATA[http://corp2.ab.myapp.com/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>
<DnsOverride>
  <set>
    </set>
  </dnsOverrides>
<urlBlacklist>
  <count>4</count>
  <list>
    <UrlEntry regex="false"> <![CDATA[http://url.blacklist.2.ab.myapp.com]]> </UrlEntry>
    <UrlEntry regex="false"> <![CDATA[http://url.blacklist.3.ab.myapp.com]]> </UrlEntry>
    <UrlEntry regex="true"> <![CDATA[http://rg.blacklist.*.ab.myapp.com]]> </UrlEntry>
    <UrlEntry regex="true"> <![CDATA[http://rg.blacklist.*?]]> </UrlEntry>
  </list>
</urlBlacklist>
<urlWhitelist>
  <count>4</count>
  <list>
    <UrlEntry regex="true"> <![CDATA[http://rg.whitelist.*.ab.myapp.com]]> </UrlEntry>
    <UrlEntry regex="true"> <![CDATA[http://rg.whitelist.*?]]> </UrlEntry>
    <UrlEntry regex="false"> <![CDATA[http://url.whitelist.2.ab.myapp.com]]> </UrlEntry>
    <UrlEntry regex="false"> <![CDATA[http://url.whitelist.3.ab.myapp.com]]> </UrlEntry>
  </list>
</urlWhitelist>
<postDataBlacklist>
  <count>2</count>
  <list>
    <UrlEntry regex="true"> <![CDATA[http://rg.postdatblacklist.*.ab.myapp.com]]> </UrlEntry>
    <UrlEntry regex="true"> <![CDATA[http://rg.postdatblacklist.*?]]> </UrlEntry>
  </list>
</postDataBlacklist>
<authRecords>
  <count>0</count>
</authRecords>
<authRecords/>
<useRobots>BLACKLIST</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>
</createdBy>
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_name]]></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_name]]></url>
      <owner>
        <id>4354</id>
        <username>acme_ss2</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>true</progressiveScanning>
      <urlBlacklist>
        <count>0</count>
      </urlBlacklist>
      <urlWhitelist>
        <count>0</count>
      </urlWhitelist>
      <postDataBlacklist>
        <count>0</count>
      </postDataBlacklist>
      <authRecords>
        <count>0</count>
      </authRecords>
    </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**

```xml
<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
<?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>
```
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 154153533111147). 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>154153533111147</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>1541535331147</id>
            <name><![CDATA[TagForScanner]]></name>
          </Tag>
        </list>
      </defaultScannerTags>
      <scannerLocked>false</scannerLocked>
      <progressiveScanning>false</progressiveScanning>
      <urlBlacklist>
        <count>0</count>
      </urlBlacklist>
      <urlWhitelist>
        <count>0</count>
      </urlWhitelist>
      <postDataBlacklist>
        <count>0</count>
      </postDataBlacklist>
      <logoutRegexList>
        <logoutRegex><![CDATA[My Web Application]]></logoutRegex>
      </logoutRegexList>
    </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**

```xml
<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>
          </SeleniumScript>
        </set>
        <requiresAuthentication>true</requiresAuthentication>
        <startingUrlRegex>true</startingUrlRegex>
      </crawlingScripts>
    </WebApp>
  </data>
</ServiceRequest>
```

XML response

```xml
<?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/xsd/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>
      <urlBlacklist>
        <count>0</count>
      </urlBlacklist>
      <urlWhitelist>
        <count>0</count>
      </urlWhitelist>
      <postDataBlacklist>
        <count>0</count>
      </postDataBlacklist>
      <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>
    </SeleniumScript>
  </list>
</crawlingScripts>

<?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>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>
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 enabled 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"?>
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
<?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>
```

**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.

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.

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**

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

**XML response**

```xml
<?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**

```xml
<?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>
```
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**

```xml
<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>
      <urlBlacklist>
        <set>
          <UrlEntry><![CDATA[http://url.blacklist.1.mywebapp.com]]></UrlEntry>
          <UrlEntry regex="false"><![CDATA[http://url.blacklist.2.mywebapp.com]]></UrlEntry>
          <UrlEntry regex="true"><![CDATA[http://rg.blacklist.*.com]]></UrlEntry>
        </set>
      </urlBlacklist>
    </WebApp>
  </data>
</ServiceRequest>
```
XML response

```xml
<?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**

```xml
<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
<?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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Web application ID.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Web application name.</td>
</tr>
<tr>
<td>url</td>
<td>(text) The URL of web application.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Tag name assigned to web application.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) Tag ID assigned to web application.</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the web application was created in WAS, in UTC date/time format.</td>
</tr>
</tbody>
</table>
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/1234"
```

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**

```bash
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**

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

**XML response**

```xml
<?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
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Web application ID.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Web application name.</td>
</tr>
<tr>
<td>url</td>
<td>(text) The URL of web application.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Tag name assigned to web application.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) Tag ID assigned to web application.</td>
</tr>
</tbody>
</table>
**Qualys Web Application Scanning API**

**Web Applications**

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

**Sample - Purge a single web application**

Let us purge the web application with ID 32420.

**API request**

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" https://qualysapi.qualys.com/qps/rest/3.0/purge/was/webapp/32420
```

**XML response**

```xml
<?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
<?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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Web application ID. This element is assigned by the service and required for an update request.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The web application name (maximum 256 characters). This element is required to create a web application.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.</td>
</tr>
<tr>
<td>url</td>
<td>(text) The URL of the web application maximum 2048 characters). This element is required to create a web application.</td>
</tr>
<tr>
<td>os</td>
<td>(text) The operating system of the web application.</td>
</tr>
<tr>
<td>owner</td>
<td>(text) This element is assigned by the service and may be specified for an update request only.</td>
</tr>
<tr>
<td>config*</td>
<td>Configure the cancel scan option. Specify “cancel after” time or “cancel at” time. Only one of <code>&lt;cancelScansAfterNHours&gt;</code> or <code>&lt;cancelScanAt&gt;</code> is allowed in one config section. Example for “cancel after” time:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;config&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;cancelScansAfterNHours&gt;3 &lt;/cancelScansAfterNHours&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/config&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Example for “cancel at” time:</td>
</tr>
</tbody>
</table>
<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.

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>
### Web Applications

**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.

**Malware Monitoring**

**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.  

Example: `<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.

```xml
<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:

```xml
<defaultScanner>
  <type>INTERNAL</type>
  <friendlyName>dp_scanner</friendlyName>
</defaultScanner>
```

### proxy.id

(integer) The default proxy for scanning the web application.

Example:

```xml
<proxy>
  <id>12345</id>
</proxy>
```

### scannerLocked

(boolean) A flag indicating whether the default scanner appliance is locked for the web application.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>&lt;scannerLocked&gt;false&lt;/scannerLocked&gt;</td>
</tr>
<tr>
<td>dnsOverrides*</td>
<td>Assign DNS override settings, one or more records, to a web application.</td>
</tr>
<tr>
<td>Example:</td>
<td>&lt;dnsOverrides&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;set&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;DnsOverride&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;id&gt;2022&lt;/id&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/DnsOverride&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/set&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/dnsOverrides&gt;</td>
</tr>
<tr>
<td>useRobots (keyword)</td>
<td>A flag indicating whether to observe the Robots.txt file and its directives if found when scanning the web application.</td>
</tr>
<tr>
<td></td>
<td>If set to IGNORE (default) the Robots.txt file is ignored.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>If set to BLACKLIST the “disallow” directives in the Robots.txt file will be observed; this means scans will not crawl matching links.</td>
</tr>
<tr>
<td>useSitemap (Boolean)</td>
<td>A flag indicating whether to adhere to a sitemap.xml file if present in the web application: true or false (default).</td>
</tr>
<tr>
<td>headers*</td>
<td>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.</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API

Web Applications

urlBlacklist* The URLs for the black 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.

urlWhitelist* The URLs for the white 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.

postDataBlacklist* The web application URLs for which you want to block 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 Application 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> ..... </data>
      <requiresAuthentication>
        true
      </requiresAuthentication>
      <startingUrl>URL</startingUrl>
      <startingUrlRegex>
        true
      </startingUrlRegex>
    </SeleniumScript>
  </list>
</crawlingScripts>
```

<table>
<thead>
<tr>
<th>Elements Assigned by the Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (integer)</td>
<td>The web application ID.</td>
</tr>
<tr>
<td>owner (text)</td>
<td>The user login ID of the web application owner.</td>
</tr>
<tr>
<td>isScheduled (boolean)</td>
<td>Is a scan scheduled for the web application?</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>createdBy</td>
<td>(text) The user who created the web application.</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the web application was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>updatedBy</td>
<td>(text) The user who last updated the web application.</td>
</tr>
<tr>
<td>updatedDate</td>
<td>(date) The date when the web application was last updated in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>lastScan</td>
<td>(text) The scan ID of the last scan run on the web application.</td>
</tr>
<tr>
<td>lastScan.status</td>
<td>(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Authentication record ID.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Authentication record name.</td>
</tr>
<tr>
<td>tags</td>
<td>(integer) Tag associated with the authentication record.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Tag name assigned to the authentication record.</td>
</tr>
</tbody>
</table>
**Qualys Web Application Scanning API**

**Authentication**

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

**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**

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

**XML response**

```xml
<?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

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

**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
<?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>
  <hasMoreRecords>false</hasMoreRecords>
</ServiceResponse>
```
<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>

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**

```bash
curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/3.0/get/was/webappauthrecord/74078"
```

**XML response**

```xml
<?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>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```
<name><![CDATA[My Authentication Record]]></name>
<owner>
  <id>4354</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>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>
<createdAt>
  <id>4354</id>
  <username>username</username>
  <firstName><![CDATA[John]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</createdAt>
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**

```bash
curl -n -u "USERNAME:PASSWORD" 
"https://qualysapi.qualys.com/qps/rest/3.0/get/was/webappauthrecord/761533"
```

**XML response**

```xml
<?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>quays_js</username>
        <firstName><![CDATA[John]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
      </owner>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```
```xml
<formRecord>
  <type>SELENIUM</type>
  <seleniumScript>
    <!--[CDATA[seleniumScript]]>
    <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>
              </tbody>
            </table>
          </body>
        </html>
      ]]>
    </data>
  </seleniumScript>
</formRecord>
```
Authentication

```xml
<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>
</WebAppAuthServerRecordField>
</fields>
</serverRecord>
```
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/761534"

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>quays_js2</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/"/>
  </head>
  <title>AuthScript</title>
  <body>
    <table cellpadding="1" cellspacing="1" border="1">
      <thead>
        <tr>
          <td rowspan="1" colspan="3">AuthScript</td>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td rowspan="1" colspan="3">AuthScript</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>]]>
    </data>
  </seleniumScript>
</formRecord>
Qualys Web Application Scanning API

Authentication

```html
<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>
</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>
```
<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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(text) Authentication record name.</td>
</tr>
<tr>
<td>WebAuthRecord</td>
<td>(text) Details associated with the web application authentication record.</td>
</tr>
<tr>
<td>tags</td>
<td>(text) Tag associated with the authentication record.</td>
</tr>
<tr>
<td>comments</td>
<td>(text) User-defined comments.</td>
</tr>
</tbody>
</table>

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**

```bash
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
<ServiceRequest>
  <data>
    <WebAppAuthRecord>
      <name><![CDATA[STANDARD auth]]></name>
      <formRecord>
        <type>STANDARD</type>
        <sslOnly>true</sslOnly>
        <fields>
          <set>
            <WebAppAuthFormRecordField>
              <name>user_john</name>
              <value>johnuser</value>
            </WebAppAuthFormRecordField>
            <WebAppAuthFormRecordField>
              <name>user_jim</name>
              <value>jimuser</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
<?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[user_john]]></name>
              <value><![CDATA[johnuser]]></value>
            </WebAppAuthFormRecordField>
            <WebAppAuthFormRecordField>
              <id>835051</id>
              <name><![CDATA[user_jim]]></name>
              <value><![CDATA[user_jim]]></value>
            </WebAppAuthFormRecordField>
          </list>
        </fields>
      </formRecord>
    </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

```xml
<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>
              <value>fake password</value>
              <secured>false</secured>
            </WebAppAuthFormRecordField>
          </set>
        </fields>
        <comments>
          <set>
            <Comment><contents><![CDATA[some comments]]></contents></Comment>
          </set>
        </comments>
      </formRecord>
    </WebAppAuthRecord>
  </data>
</ServiceRequest>
```

XML response

```xml
<?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>
</ServiceResponse>
```
<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>
    </comments>
  </WebAppAuthRecord>
</data>
Sample - Create a Selenium script

**API request**

```bash
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>
Qualys Web Application Scanning API
Authentication

XML response

```xml
<?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>acme_as2</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">
```
```html
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="https://community.qualys.com/" />
<title>seleniumScriptOK</title>
<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>```
Qualys Web Application Scanning API
Authentication

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>

```

```
</tags>
</comments>
</count>0</count>
</comments>
<createdDate>2017-05-06T16:23:43Z</createdDate>
<createdBy>
  <id>4354</id>
  <username>acme_as2</username>
  <firstName>![CDATA[Alice]]>
</firstName>
  <lastName>![CDATA[Smith]]>
</lastName>
</createdBy>
<updatedDate>2017-05-06T16:23:43Z</updatedDate>
<updatedBy>
  <id>4354</id>
  <username>acme_as2</username>
  <firstName>![CDATA[Alice]]>
</firstName>
  <lastName>![CDATA[Smith]]>
</lastName>
</updatedBy>
</WebAppAuthRecord>
</data>
</ServiceResponse>
<name><![CDATA[server auth]]></name>
<serverRecord>
  <sslOnly>true</sslOnly>
  <certificate>
    <name><![CDATA[My Certificate]]></name>
    <contents><![CDATA[-----BEGIN CERTIFICATE-----
MIIC4jCCAkugAwIBAgIWKw6GX2aMMA0GCSqGSIb3DQEBBQUAMIGJMQswCQYD
VQQGEwJGUjEPMA0GA1UECAwGRnJhbmNlMRQwDwYDVQQHEhNMREwDQYJKoZIhvc
... 
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC4Si/HaNxQtwQUtot867MxTP1PqAqh7VyiHIdBso37eafpdB6
aphhi07wo2rZrcWniUUhwpvlL4apG470/RzkIKSu4h9akHqA5b0Pe0ZasrE7B
MxUZWNf9dfrYJXQmDcPc1i4w4zZt9+PabXMy5Mg9ONEUKS3AONChK7acwIDAQAB
AoGAMHwAFLFdgLzQXNMPZ6uGv4TaaJkzT2YEzKLIyv7e///Dt160GwDSpH3Lqffh
... 
-----END RSA PRIVATE KEY-----]]></contents>
    <passphrase>My Certificate</passphrase>
  </certificate>
  <comments>
    <set>
      <Comment><contents><![CDATA[some comments]]></contents></Comment>
    </set>
  </comments>
</serverRecord>
<comments/>
<WebAppAuthRecord>
  <id>685134</id>
</WebAppAuthRecord>
</ServiceRequest>

**XML response**

```xml
<?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>
    </WebAppAuthRecord>
  </data>
</ServiceResponse>
```
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

Sample - Update authentication record settings

Let us update the settings for authentication record ID 82605.

API request

```
```

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>
      </serverRecord>
      <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">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <WebAppAuthRecord>
            <id>82605</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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Authentication record ID.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Authentication record name.</td>
</tr>
<tr>
<td>tags</td>
<td>(integer) Tag associated with the authentication record.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Tag name assigned to the authentication record.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) Tag ID assigned to the authentication record.</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the authentication record</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API
Authentication

was created in WAS, in UTC date/time format.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>updatedDate</td>
<td>(date) The date when the authentication record was updated in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>lastScan.date</td>
<td>(date) The date when the web application (associated with the authentication record) was last scanned, in UTC date/time format.</td>
</tr>
<tr>
<td>lastScan.authStatus</td>
<td>(keyword) Authentication status reported by the last web application scan: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL</td>
</tr>
<tr>
<td>isUsed</td>
<td>(boolean) Indicates whether used by a web application or scan.</td>
</tr>
<tr>
<td>contents</td>
<td>(keyword) : FORM_STANDARD, FORM_CUSTOM, FORM_SELENIUM, SERVER_BASIC, SERVER_DIGEST</td>
</tr>
</tbody>
</table>

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
<?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**
Qualys Web Application Scanning API
Authentication

<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.

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

<table>
<thead>
<tr>
<th>FORM_SELENIUM, SERVER_BASIC, SERVER_DIGEST</th>
</tr>
</thead>
</table>

**WebAuthRecord** (text) Details associated with the web application authentication record.

**comments** (text) User-defined comments.
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The scan ID.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The scan name.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) The name of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) The ID of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.tags</td>
<td>(integer) The tags associated with the web application being scanned.</td>
</tr>
<tr>
<td>webApp.tags.id</td>
<td>(integer) The tag ID assigned to web application being scanned.</td>
</tr>
</tbody>
</table>
## Qualys Web Application Scanning API
### Scans

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>(text) Scan Reference ID.</td>
</tr>
<tr>
<td>launchedDate</td>
<td>(date) The date and time when the scan was launched in UTC date/time format (YYYY-MM-DDT HH:MM:SSZ).</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td>mode</td>
<td>(keyword) The mode of the scan: ONDEMAND, SCHEDULED or API.</td>
</tr>
<tr>
<td>status</td>
<td>(keyword) The status of the scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED.</td>
</tr>
<tr>
<td>authStatus</td>
<td>(Keyword) Indicates the status of the authentication record: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL.</td>
</tr>
<tr>
<td>resultsStatus</td>
<td>(keyword) The status of the scan: NOT_USED, NO_HOST_ALIVE, NO_WEB_SERVICE, PROCESSING, SCAN_RESULTS_INVALID, TIME_LIMIT_REACHED, SERVICE_ERROR, SCAN_INTERNAL_ERROR, SUCCESSFUL, TO_BE_PROCESSED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, SUBMITTED, RUNNING, FINISHED, CANCELED, CANCELING ERROR, DELETED.</td>
</tr>
</tbody>
</table>

### Sample - Get count of scans in user’s account

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

**API request**

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

**XML response**

```xml
<?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>
</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"?>
  <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

```bash
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

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

XML response

```xml
<?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

```bash
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

```xml
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags.id" operator="EQUALS">1516928</Criteria>
  </filters>
</ServiceRequest>
```
Qualys Web Application Scanning API

Scans

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/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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The scan ID.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The scan name.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) The name of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) The ID of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.tags</td>
<td>(integer) The tags associated with the web application being scanned.</td>
</tr>
<tr>
<td>webApp.tags.id</td>
<td>(integer) The tag ID assigned to web application being scanned.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>(text) Scan Reference ID.</td>
</tr>
<tr>
<td>launchedDate</td>
<td>(date) The date and time when the scan was launched in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td>mode</td>
<td>(keyword) The mode of the scan: ONDEMAND, SCHEDULED or API.</td>
</tr>
<tr>
<td>status</td>
<td>(keyword) The status of the scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED.</td>
</tr>
<tr>
<td>authStatus</td>
<td>(keyword) Indicates the status of the authentication record: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL.</td>
</tr>
<tr>
<td>resultsStatus</td>
<td>(keyword) The status of the scan: NOT_USED, NO_HOST_ALIVE, NO_WEB_SERVICE, PROCESSING, SCAN_RESULTS_INVALID, TIME_LIMIT_REACHED, SERVICE_ERROR, SCAN_INTERNAL_ERROR, SUCCESSFUL, TO_BE_PROCESSED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, SUBMITTED, RUNNING, FINISHED, CANCELED, CANCELING ERROR, DELETED.</td>
</tr>
</tbody>
</table>

**Sample - List running scans**

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

**API request**

```bash
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>
```

143
<Criteria field="status" operator="EQUALS">RUNNING</Criteria>
</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>
    </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>
    </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**

```bash
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**

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

**XML response**

```xml
<?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>
```
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

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

XML response

```xml
<?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>
        </scannerAppliance>
      </target>
    </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**

```xml
<ServiceRequest>
  <friendlyName>
    <![CDATA[John_doe]]>
  </friendlyName>
  <scannerAppliance>
    <cancelOption>SPECIFIC</cancelOption>
    <randomizeScan>false</randomizeScan>
  </scannerAppliance>
  <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>
  </WasScan>
</data>
</ServiceResponse>
```
<filters>
    <Criteria field="webApp.tags.id" operator="EQUALS">8158322</Criteria>
</filters>

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/wasscan.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <hasMoreRecords>false</hasMoreRecords>
    <data>
        <WasScan>
            <id>2208317</id>
            <name>1538976557822_Scan16</name>
            <reference>was/1538976670564.372113</reference>
            <type>VULNERABILITY</type>
            <mode>API</mode>
            <multi>false</multi>
            <target>
                <webApp>
                    <id>1472824</id>
                    <name>web app 1538976530195</name>
                    <url>http://10.11.72.39</url>
                </webApp>
                <scannerAppliance>
                    <type>INTERNAL</type>
                    <friendlyName>John_doe</friendlyName>
                </scannerAppliance>
            </target>
            <cancelOption>SPECIFIC</cancelOption>
            <randomizeScan>false</randomizeScan>
        </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
<?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>
      </target>
      <profile>
        <id>28147</id>
        <name>
          <![CDATA[My Option Profile]]>
        </name>
      </profile>
    </WasScan>
  </data>
</ServiceResponse>
```
<launchDate>2017-11-01T20:12:23Z</launchDate>
<launchedBy>
  <id>2226741</id>
  <username>acme_ak1</username>
  <firstName><![CDATA[Amy]]></firstName>
  <lastName><![CDATA[Kim]]></lastName>
</launchedBy>
<status>CANCELED</status>
<canceledBy>
  <id>9872437571</id>
  <username>acme_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
<?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>
    </WasScan>
  </data>
</ServiceResponse>
```
<name>127.0.0.1</name>
</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>
    </WasScanOption>
  </list>
</options>
Qualys Web Application Scanning API

Scans

<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>acme_ak1</username>
  <firstName><![CDATA[Amy]]></firstName>
  <lastName><![CDATA[Kim]]></lastName>
</launchedBy>
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
<?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>
        </webApp>
      </target>
    </WasScan>
  </data>
</ServiceResponse>
```
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**

```xml
<ServiceResponse
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/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>true</progressiveScanning>
    </WasScan>
  </data>
</ServiceResponse>
```
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(text) The scan name.</td>
</tr>
<tr>
<td>webApps.id or tags.id</td>
<td>(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</td>
</tr>
</tbody>
</table>
### Scans

**Qualys Web Application Scanning API**

**Scans**

- **type**
  - `(keyword)` The scan type: VULNERABILITY or DISCOVERY.

- **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:
  ```xml
  <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
Qualys Web Application Scanning API

Scans

authentication record for the scan.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>proxy.id</td>
<td>(integer) The proxy for scanning the target web application.</td>
</tr>
<tr>
<td>Example:</td>
<td>&lt;proxy&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;id&gt;12345&lt;/id&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/proxy&gt;</td>
</tr>
<tr>
<td>dnsOverride.id</td>
<td>(integer) The DNS override record for scanning the target web application.</td>
</tr>
<tr>
<td>Example:</td>
<td>&lt;dnsOverride&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;id&gt;67890&lt;/id&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/dnsOverride&gt;</td>
</tr>
<tr>
<td>sendMail</td>
<td>(boolean) Set to false to disable scan complete email notifications.</td>
</tr>
<tr>
<td>Example:</td>
<td>&lt;sendMail&gt;false&lt;/sendMail&gt;</td>
</tr>
</tbody>
</table>

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/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/148422839357.1955345</reference>
      <type>DISCOVERY</type>
      <mode>ONDEMAND</mode>
      <progressiveScanning>true</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>
        </list>
      </options>
    </WasScan>
  </data>
<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>
<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>
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>false</progressiveScanning>
    </WasScan>
  </data>
</ServiceRequest>
XML response

```xml
<?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
<?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

```xml
<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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(text) The scan name.</td>
</tr>
<tr>
<td>target.webApp.id1</td>
<td>(integer) The ID of the web application being scanned.</td>
</tr>
<tr>
<td>target.tags.excluded.option</td>
<td>(keyword: ALL or ANY) Decides which web applications should be excluded from the scan.</td>
</tr>
<tr>
<td></td>
<td>ALL : Only the web applications</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>target.tags.excluded.tagList.Tag.id</td>
<td>(integer) The web applications associated with the tag (identified by the specified tag ID) are excluded from the scan.</td>
</tr>
<tr>
<td>target.tags.included.option</td>
<td>(keyword: ALL or ANY) Decides which web applications should be included in the scan.</td>
</tr>
<tr>
<td>target.tags.included.tagList.Tag.id</td>
<td>(integer) The web applications associated with the tag (identified by the specified tag ID) are included in the scan.</td>
</tr>
<tr>
<td>options</td>
<td>(keyword: ANY, ALL) Decides which web applications should be included or excluded from the scan.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword: EXTERNAL or INTERNAL</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API

Scans

or scannerTags) Type of the scanner appliance to be used for the scan.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| profile.id2              | (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. |
Qualys Web Application Scanning API

Scans

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.

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>
  </WasScan>
</data>
</ServiceRequest>
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
<?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>
```
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

```xml
<ServiceRequest>
<data>
  <WasScan>
    <name><![CDATA[sample Scan]]></name>
    <type>VULNERABILITY</type>
    <target>
      <webApps>
        <set>
          <WebApp>
            <id>2376280</id>
          </WebApp>
        </set>
      </webApps>
    </target>
  </WasScan>
</data>
</ServiceRequest>
```
<table>
<thead>
<tr>
<th>WebApp</th>
<th>&lt;id&gt;4114251&lt;/id&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>set</td>
<td></td>
</tr>
<tr>
<td>webApps</td>
<td></td>
</tr>
<tr>
<td>scannerAppliance</td>
<td>type=EXTERNAL</td>
</tr>
<tr>
<td>cancelOption</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>target</td>
<td></td>
</tr>
<tr>
<td>profile</td>
<td>&lt;id&gt;2231014&lt;/id&gt;</td>
</tr>
<tr>
<td>WasScan</td>
<td></td>
</tr>
<tr>
<td>data</td>
<td></td>
</tr>
<tr>
<td>ServiceRequest</td>
<td></td>
</tr>
</tbody>
</table>

XML response

```xml
<?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>1275177</id>
    </WasScan>
  </data>
</ServiceResponse>
```

XSD

```xml
<platform_API_server>/qps/xsd/3.0/was/wasscan.xsd
```
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

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @- "https://qualysapi.qualys.com/qps/rest/3.0/status/was/wasscan/1902350"
```

XML response

```xml
<?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>
      <summary>
```
Scans

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]

Retrieves 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.

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 -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-"https://qualysapi.qualys.com/qps/rest/3.0/download/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_quays_tc321]]></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>
    </list>
  </options>
</WasScan>
Qualys Web Application Scanning API

Scans

<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>username</username>
  <firstName><![CDATA[John]]></firstName>
  <lastName><![CDATA[Smith]]></lastName>
</launchedBy>

<status>FINISHED</status>
<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>
  </global>
</stats>
<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>0</nbLevel3>
      <nbLevel2>14</nbLevel2>
      <nbLevel1>0</nbLevel1>
    </WascStat>
    ...
  </list>
</byWasc>
Qualys Web Application Scanning API

Scans

...
</instances>
</WasScanVuln>
...
</list>
</vulns>
<sensitiveContents>
<count>0</count>
</sensitiveContents>
<igs>
<count>10</count>
</list>
</WasScanIg>

<XSD>
<platform_API_server>/qps/xsd/3.0/was/wasscan.xsd
</XSD>
Cancel Scan

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

[GET]

Cancel an unfinished scan on a web application which is in the user's scope. Note that scan results will not be returned.

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

```bash
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
<?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>
```
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The scan ID.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The scan name.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) The name of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) The ID of the web application being scanned.</td>
</tr>
<tr>
<td>reference</td>
<td>(text) Scan Reference ID.</td>
</tr>
<tr>
<td>launchedDate</td>
<td>(date) The date and time when the scan was launched in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API

**Scans**

<table>
<thead>
<tr>
<th><strong>type</strong></th>
<th>(keyword) The scan type: VULNERABILITY or DISCOVERY.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mode</strong></td>
<td>(keyword) The mode of the scan: ONDEMAND, SCHEDULED or API.</td>
</tr>
<tr>
<td><strong>status</strong></td>
<td>(keyword) The status of the scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED.</td>
</tr>
<tr>
<td><strong>authStatus</strong></td>
<td>(Keyword) Indicates the status of the authentication record: NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL.</td>
</tr>
<tr>
<td><strong>resultsStatus</strong></td>
<td>(keyword) The status of the scan: NOT_USED, NO_HOST_ALIVE, NO_WEB_SERVICE, PROCESSING, SCAN_RESULTS_INVALID, TIME_LIMIT_REACHED, SERVICE_ERROR, SCAN_INTERNAL_ERROR, SUCCESSFUL, TO_BE_PROCEEDED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, SUBMITTED, RUNNING, FINISHED, CANCELED, CANCELING, ERROR, DELETED.</td>
</tr>
</tbody>
</table>

**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
<?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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(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).</td>
</tr>
<tr>
<td>name</td>
<td>(text) The user-defined scan name (maximum 256 characters).</td>
</tr>
<tr>
<td>target* (for single web application)</td>
<td>(text) The target of the scan. The target includes the web application and authentication records, if any.</td>
</tr>
</tbody>
</table>

```
<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>
</target>
```
Qualys Web Application Scanning API

Scans

</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 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> (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>
### Qualys Web Application Scanning API

#### Scans

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>type</strong></td>
<td>(keyword) The scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td><strong>sendMail</strong></td>
<td>(boolean) Set to false to disable scan complete email notifications.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>&lt;sendMail&gt;false&lt;/sendMail&gt;</code></td>
</tr>
<tr>
<td><strong>profile.id</strong></td>
<td>(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.</td>
</tr>
</tbody>
</table>
| 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>`                                                 |
| **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. |
### Qualys Web Application Scanning API

#### Scans

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>status</strong></td>
<td>(keyword) The status of the scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR or CANCELED.</td>
</tr>
<tr>
<td><strong>endScanDate</strong></td>
<td>(date) The date and time when the scan ended in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
<tr>
<td><strong>summary</strong></td>
<td>The scan summary. <code>&lt;crawlTime&gt;</code> is the length of time used to crawl the web application. <code>&lt;testDuration&gt;</code> is the length of time used to perform analysis. <code>&lt;nbRequests&gt;</code> is the number of requests sent during the scan. <code>&lt;authStatus&gt;</code> is the authentication status (NONE, NOT_USED, SUCCESSFUL, FAILED or PARTIAL)</td>
</tr>
<tr>
<td><strong>vulns</strong></td>
<td>The list of detected vulnerabilities. Each <code>&lt;WasScanVuln&gt;</code> element identifies a particular vulnerability QID and the URI where detected, each <code>&lt;WasScanVulnInstance&gt;</code> element identifies a vulnerability instance, and each <code>&lt;WasScanVulnInstancePayload&gt;</code> element identifies associated payloads.</td>
</tr>
<tr>
<td><strong>igs</strong></td>
<td>The detected information gathered. Each <code>&lt;WasScanIg&gt;</code> element identifies a particular information gathered QID.</td>
</tr>
<tr>
<td><strong>sensitiveContents</strong></td>
<td>The detected sensitive content. Each <code>&lt;WasScanSensitiveContent&gt;</code> element identifies the sensitive content.</td>
</tr>
</tbody>
</table>
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.
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
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT WEB_APPLICATION_SCAN (ERROR | (HEADER, SUMMARY, RESULTS))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>
<!ELEMENT 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)>
<!ELEMENT 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)>
```
Qualys Web Application Scanning API

Scans

<!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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The user-defined schedule name (maximum 256 characters).</td>
</tr>
<tr>
<td>owner.id</td>
<td>(integer) ID associated with the owner who created the schedule.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>updatedDate</td>
<td>(date) The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) The name of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) The ID of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.tags</td>
<td>Tags associated with the web application being scanned.</td>
</tr>
<tr>
<td></td>
<td>(with operator=&quot;NONE&quot;)</td>
</tr>
<tr>
<td>webApp.tags.id</td>
<td>(integer) ID of the tag applied to the web application being scanned.</td>
</tr>
<tr>
<td>invalid</td>
<td>(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.</td>
</tr>
<tr>
<td>active</td>
<td>(boolean) Indicates whether the schedule is active or not. True indicates active schedule.</td>
</tr>
</tbody>
</table>

**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**
Qualys Web Application Scanning API

Schedules

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**

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

**XML response**

```xml
<?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

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

XML response

```xml
<?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

```xml
<ServiceRequest>
  <filters>
    <Criteria field="webApp.tags.id" operator="EQUALS">1516928</Criteria>
  </filters>
</ServiceRequest>
```
Qualys Web Application Scanning API

Schedules

XML response

```xml
<?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

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>(boolean) Indicates whether the schedule is active or not. True indicates active schedule.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) The name of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) The ID of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.tags</td>
<td>Tags associated with the web application being scanned.</td>
</tr>
<tr>
<td>webApp.tags.id</td>
<td>(integer) ID of the tag applied to the web application being scanned.</td>
</tr>
<tr>
<td>invalid</td>
<td>(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.</td>
</tr>
<tr>
<td>multi</td>
<td>(boolean) Indicates if the scheduled scan is single scan or multiple scan.</td>
</tr>
<tr>
<td>lastScan</td>
<td>(boolean) Indicates if the last scan was performed or not. True indicates that the last scan was performed.</td>
</tr>
<tr>
<td>lastScan.launchedDate</td>
<td>(date) Date when the last scan was launched on the web application, in UTC date/time format.</td>
</tr>
<tr>
<td>lastScan.status</td>
<td>(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR, CANCELED)</td>
</tr>
</tbody>
</table>
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

```bash
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

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

XML response

```xml
<?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>
      </target>
    </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.comqps/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>
  </filters>
</ServiceRequest>
```
<Criteria field="lastScan.launchedDate" operator="LESSER">2017-08-19</Criteria>
</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>
```
Qualys Web Application Scanning API

Schedules

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/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/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>
  </owner>
Sample - List active schedules
Let us view a list of all schedules that are in the user’s scope and were launched.

**API request**

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

**Request POST data**

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

**XML response**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="https://qualysapi.p04.eng.sjc01.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>
```

212
<id>8077389</id>
<name>
<![CDATA[SampleWebApp_1538665472012 ]]> 
</name>
<url>
<![CDATA[http://funkytown.vuln.qa.afco.com:80/cassium/xss/]]>
</url>
</webApp>
<scannerAppliance>
  <type>EXTERNAL</type>
</scannerAppliance>
<cancelOption>SPECIFIC</cancelOption>
</target>
PROFILE>
</profile>
<scheduling>
  <startDate>2018-10-08T16:41:00Z</startDate>
  <timeZone>
    <code>Asia/Colombo</code>
    <offset>+05:30</offset>
  </timeZone>
  <occurrenceType>ONCE</occurrenceType>
  <nextLaunchDate>2018-10-09T11:11:00Z</nextLaunchDate>
  <createdAt>2018-10-08T11:12:28Z</createdAt>
  <updatedAt>2018-10-08T11:12:29Z</updatedAt>
</scheduling>
</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/714393"
```

**XML response**

```
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>714393</id>
      <name>
```
Qualys Web Application Scanning API

Schedules

<![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>
                <Tag>
                    <id>12075819</id>
                    <name><![CDATA[New_tag]]></name>
                </Tag>
                <Tag>
                    <id>2685657</id>
                    <name><![CDATA[Business Units]]></name>
                </Tag>
            </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>
Sample - View schedule details (progressive scan)
The progressiveScanning element will be included in the call response, if Progressive Scanning is enabled for the subscription.

<table>
<thead>
<tr>
<th>API request</th>
</tr>
</thead>
<tbody>
<tr>
<td>curl -n -u &quot;USERNAME:PASSWORD&quot; &quot;<a href="https://qualysapi.qualys.com/qps/rest/3.0/get/wasscanschedule/8183">https://qualysapi.qualys.com/qps/rest/3.0/get/wasscanschedule/8183</a>&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XML response</th>
</tr>
</thead>
</table>
| <?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/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/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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(text) Name of the schedule.</td>
</tr>
<tr>
<td>target.webApp.id1</td>
<td>(integer) The web applications to be scanned.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td>profile.id2</td>
<td>(integer) The name of the option profile that includes scan settings.</td>
</tr>
</tbody>
</table>

The service provides the profile “Initial WAS Options” and we recommend this to get started.

Example:
### Qualys Web Application Scanning API

#### Schedules

```xml
<profile>
  <name>Initial WAS Options</name>
</profile>
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startDate</td>
<td>(date) The date when the schedule starts in UTC date/time format.</td>
</tr>
<tr>
<td>timeZone</td>
<td>(text) The timezone in which the scan is scheduled in UTC date/time format.</td>
</tr>
<tr>
<td>occurrenceType</td>
<td>(keyword) The frequency of the scheduled scan: ONCE, DAILY, WEEKLY or MONTHLY.</td>
</tr>
<tr>
<td>notification</td>
<td>(boolean) A flag indicating whether email notification is enabled for scheduled scan.</td>
</tr>
<tr>
<td>reschedule</td>
<td>(boolean) Set this flag to reschedule the scan.</td>
</tr>
<tr>
<td>target.scannerAppliance.type</td>
<td>(keyword) The type of scanner appliance used for the scan: EXTERNAL or INTERNAL or scannerTags.</td>
</tr>
<tr>
<td>target.scannerAppliance.friendlyName</td>
<td>(text) Name of the scanner appliance used for the scan.</td>
</tr>
<tr>
<td>target.scannerTags.set.Tag.id</td>
<td>(integer) The scanner associated with the tag (identified by the specified tag ID) is picked for the scan.</td>
</tr>
<tr>
<td>target.webAppAuthRecord.id or target.webAppAuthRecord.isDefault</td>
<td>Decides the authentication record to be used for the scan.</td>
</tr>
</tbody>
</table>

`target.webAppAuthRecord.id` (integer): Specify the web
### Schedules

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>application's authentication record ID</code></td>
<td>To use the specific authentication record.</td>
</tr>
<tr>
<td><code>target.webAppAuthRecord.isDefault boolean</code></td>
<td>Set to true to use the default web application's authentication record for the scan.</td>
</tr>
<tr>
<td><code>options (keyword: ANY, ALL)</code></td>
<td>Decides which web applications should be excluded from the scan.</td>
</tr>
<tr>
<td>ALL</td>
<td>Only the web applications associated with all the specified tags are excluded from the scan.</td>
</tr>
<tr>
<td>ANY</td>
<td>Only the web applications associated with any of the specified tags are excluded from the scan.</td>
</tr>
<tr>
<td><code>proxy.id (integer)</code></td>
<td>The proxy for scanning the target web application.</td>
</tr>
<tr>
<td>Example</td>
<td><code>&lt;proxy&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;id&gt;12345&lt;/id&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/proxy&gt;</code></td>
</tr>
<tr>
<td><code>dnsOverride.id (integer)</code></td>
<td>The DNS override record for scanning the target web application.</td>
</tr>
<tr>
<td>Example</td>
<td><code>&lt;dnsOverride&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;id&gt;67890&lt;/id&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/dnsOverride&gt;</code></td>
</tr>
<tr>
<td><code>cancelOption (keyword: DEFAULT, SPECIFIC)</code></td>
<td>Set to DEFAULT - Forces the use of the target web app’s cancelScans option if set, else fall back to the one</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API

Schedules

| passsed 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>` |

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**

```xml
<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>
```

221
<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"...>
<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>quays_cp</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>
  </WasScanSchedule>
</data>
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"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>325624</id>
      <name><![CDATA[My Scan Schedule]]></name>
      <owner>
        <id>2086786</id>
        <username>acme_tp16</username>
      </owner>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```
<firstName><![CDATA[FIRSTNAME]]></firstName>
lastName><![CDATA[LASTNAME]]></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>
ocurrenceType>ONCE</ocurrenceType>
cancelAfterNHours>7</cancelAfterNHours>
</scheduling>
notification>
  active>false</active>
</notification>
nextLaunchDate>2017-09-30T20:11:00Z</nextLaunchDate>
lunchedCount>0</lunchedCount>
createdDate>2017-06-26T20:54:30Z</createdDate>
 createdBy>
  <id>2086786</id>
  <username>ACME_tp16</username>
  <firstName><![CDATA[FIRSTNAME]]></firstName>
  <lastName><![CDATA[LASTNAME]]></lastName>
</createdBy>
updatedDate>2017-06-26T20:54:30Z</updatedDate>
updatedBy>
  <id>2086786</id>
</updatedBy>
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.

```
<username>quays_tp16</username>
<firstName><![CDATA[FIRSTNAME]]></firstName>
<lastName><![CDATA[LASTNAME]]></lastName>
</updatedBy>
<sendMail>true</sendMail>
</WasScanSchedule>
</data>
</ServiceResponse>
```

**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>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```
<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>
</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>
      <id>1688</id>
      <progressiveScanning>ENABLED</progressiveScanning>
    </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/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
      <id>1688</id>
    </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
<?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/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/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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(text) Name of the schedule.</td>
</tr>
<tr>
<td>webApps.id or tags.id</td>
<td>(integer) The web applications to be scanned.</td>
</tr>
<tr>
<td>webApps.id: Specify the web application ID to include it in the scan.</td>
<td></td>
</tr>
<tr>
<td>tags.id: Specify the tag ID associated</td>
<td></td>
</tr>
</tbody>
</table>
with the web applications to be scanned.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target.tags.excluded.option</td>
<td>(keyword: ALL or ANY) Decides which web applications should be excluded from the scan.</td>
</tr>
<tr>
<td></td>
<td>ALL: Only the web applications associated with all the specified tags are excluded from the scan.</td>
</tr>
<tr>
<td></td>
<td>ANY: Only the web applications associated with any of the specified tags are excluded from the scan.</td>
</tr>
<tr>
<td>target.tags.excluded.tagList.Tag.id</td>
<td>(integer) The web applications associated with the tag (identified by the specified tag ID) are excluded from the scan.</td>
</tr>
<tr>
<td>target.tags.included.option</td>
<td>(keyword: ALL or ANY) Decides which web applications should be included in the scan.</td>
</tr>
<tr>
<td></td>
<td>ALL: Only the web applications associated with all the specified tags are included in the scan.</td>
</tr>
<tr>
<td></td>
<td>ANY: Only the web applications associated with any of the specified tags are included in the scan.</td>
</tr>
<tr>
<td>target.tags.included.tagList.Tag.id</td>
<td>(integer) The web applications associated with the tag (identified by the specified tag ID) are included in the scan.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td>profile.id (integer)</td>
<td>(integer) The name of the option profile that includes scan settings.</td>
</tr>
</tbody>
</table>
### Qualys Web Application Scanning API

#### Schedules

The service provides the profile “Initial WAS Options” and we recommend this to get started.

Example:

```xml
<profile>
  <name>Initial WAS Options</name>
</profile>
```

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startDate (date)</td>
<td>(date) The date when the schedule starts in UTC date/time format.</td>
</tr>
<tr>
<td>timeZone (text)</td>
<td>(text) The timezone in which the scan is scheduled in UTC date/time format.</td>
</tr>
<tr>
<td>occurrenceType</td>
<td>(keyword) The frequency of the scheduled scan: ONCE, DAILY, WEEKLY or MONTHLY.</td>
</tr>
<tr>
<td>notification</td>
<td>(boolean) A flag indicating whether email notification is enabled for scheduled scan.</td>
</tr>
<tr>
<td>reschedule</td>
<td>(boolean) Set this flag to reschedule the scan.</td>
</tr>
<tr>
<td>target.authRecordOption</td>
<td>(integer) Defines the authentication record to be used during the scan.</td>
</tr>
<tr>
<td></td>
<td>Set to SPECIFIC - Always use the authRecord passed while launching the scan.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>target.profileOption</td>
<td>(keyword: ALL or ANY) Defines the option profile to be used during the scan.</td>
</tr>
</tbody>
</table>
### Schedules

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target.scannerOption</td>
<td>(integer) Defines the scanner appliance to be used during the scan.</td>
</tr>
<tr>
<td></td>
<td>Set to SPECIFIC - Always use the scanner passed while launching the scan.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>target.randomizeScan</td>
<td>Allows the service to scan the selected web applications in random order. The randomness will help prevent network slowdowns and/or errors</td>
</tr>
<tr>
<td>target.scannerAppliance.type</td>
<td>(keyword: EXTERNAL or INTERNAL or scannerTags) Type of the scanner appliance to be used for the scan.</td>
</tr>
<tr>
<td>target.scannerAppliance.friendlyName</td>
<td>(text) Name of the scanner appliance being used for the scan.</td>
</tr>
<tr>
<td>cancelOption</td>
<td>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.</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API

Schedules

set to SPECIFIC - Always use the cancel scan option passed while launching the scan.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sendMail</td>
<td>(boolean) Set to false to disable scan complete email notifications.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>&lt;sendMail&gt;false&lt;/sendMail&gt;</code></td>
</tr>
</tbody>
</table>

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
<?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>
      </scheduling>
    </WasScanSchedule>
  </data>
</ServiceRequest>
```
Qualys Web Application Scanning API

Schedules

<occurrence>
  <weeklyOccurrence>
    <everyNWeeks>2</everyNWeeks>
    <occurrenceCount>20</occurrenceCount>
    <onDays>
      <WeekDay>SATURDAY</WeekDay>
    </onDays>
  </weeklyOccurrence>
</occurrence>

<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>
Qualys Web Application Scanning API

Schedules

XML response

```xml
<?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.afco.com/cassium/xss/]]></url>
                        </WebApp>
                        <WebApp>
                            <id>4331924</id>
                            <name><![CDATA[web app 1497351100446]]></name>
                            <url><![CDATA[http://www.afco.com/cassium/xss/]]></url>
                        </WebApp>
                    </list>
                </webApps>
            </target>
        </WasScanSchedule>
    </data>
</ServiceResponse>
```
<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>
</notification>
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.

**API request**

```bash
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
<?xml version="1.0" encoding="UTF-8"?>
<ServiceRequest>
  <data>
    <WasScanSchedule>
      <name>SampleSchedule</name>
    </WasScanSchedule>
  </data>
</ServiceRequest>
```
Schedules

<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>
  </tags>
</target>
<excluded>
  <option>ANY</option>
  <tagList>
    <set>
      <Tag>
        <id>12017228</id>
      </Tag>
    </set>
  </tagList>
</excluded>
<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>
</ServiceResponse>

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>
    </WasScanSchedule>
  </data>
</ServiceResponse>
Qualys Web Application Scanning API

Schedules

```xml
<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 Server149735111801]]></name>
  </webAppAuthRecord>
</target>
<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>
Qualys Web Application Scanning API

Schedules

<XSD>
<platform API server>/qps/xsd/3.0/wasscanschedule.xsd
</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>
      </notification>
      <delay>
        <nb>4</nb>
      </delay>
    </WasScanSchedule>
  </data>
</ServiceRequest>
```
<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>
</ServiceResponse>

Sample - Update notification to reschedule

API request

```bash
```

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

Request POST data

```xml
<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>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The user-defined schedule name (maximum 256 characters).</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
</tbody>
</table>
**Qualys Web Application Scanning API**

**Schedules**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>updatedDate</td>
<td>(date) The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) The name of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) The ID of the web application being scanned.</td>
</tr>
<tr>
<td>owner.id</td>
<td>(text) ID associated with the owner who created the schedule.</td>
</tr>
<tr>
<td>active</td>
<td>(boolean) Indicates whether the schedule is active or not. True indicates active schedule.</td>
</tr>
<tr>
<td>invalid</td>
<td>(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.</td>
</tr>
</tbody>
</table>

**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/wasscanschedule/1688" < file.xml
```

**XML response**

```xml
<?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>
      <!-- Data here -->
    </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/wasscanschedule" < 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"?>
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <data>
    <WasScanSchedule>
      <id>701147</id>
    </WasScanSchedule>
    <WasScanSchedule>
      <id>701946</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>

XSD
Schedules

<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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The user-defined schedule name (maximum 256 characters).</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>updatedDate</td>
<td>(date)</td>
</tr>
<tr>
<td>type</td>
<td>(keyword)</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer)</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text)</td>
</tr>
<tr>
<td>owner.id</td>
<td>(integer)</td>
</tr>
<tr>
<td>active</td>
<td>(boolean)</td>
</tr>
<tr>
<td>invalid</td>
<td>(boolean)</td>
</tr>
</tbody>
</table>

**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/wasscanschedule/1688" < file.xml
```

**XML response**

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <WasScanSchedule>
```

---

254
Sample - Deactivate Multi Schedule using filters

**API request**

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

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

**Request POST data**

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

**XML response**

```xml
<?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/wasscanschedule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>2</count>
  <data>
    <WasScanSchedule>
      <id>701147</id>
    </WasScanSchedule>
    <WasScanSchedule>
      <id>701946</id>
    </WasScanSchedule>
  </data>
</ServiceResponse>
```

**XSD**
Qualys Web Application Scanning API

Schedules

<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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The user-defined schedule name (maximum 256 characters).</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>updatedDate</td>
<td>(date) The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
</tbody>
</table>
## Qualys Web Application Scanning API
### Schedules

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) The name of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) The ID of the web application being scanned.</td>
</tr>
<tr>
<td>owner.id</td>
<td>(integer) ID associated with the owner who created the schedule.</td>
</tr>
<tr>
<td>active</td>
<td>(boolean) Indicates whether the schedule is active or not. True indicates active schedule.</td>
</tr>
<tr>
<td>invalid</td>
<td>(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.</td>
</tr>
</tbody>
</table>

### Sample - Delete single schedule

#### API request
```bash
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
<?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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The user-defined schedule name (maximum 256 characters).</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
</tbody>
</table>
### Schedules

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>updatedDate</td>
<td>(date)</td>
<td>The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword)</td>
<td>The scheduled scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text)</td>
<td>The name of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer)</td>
<td>The ID of the web application being scanned.</td>
</tr>
<tr>
<td>owner.id</td>
<td>(integer)</td>
<td>ID associated with the owner who created the schedule.</td>
</tr>
<tr>
<td>active</td>
<td>(boolean)</td>
<td>Indicates whether the schedule is active or not. True indicates active schedule.</td>
</tr>
<tr>
<td>invalid</td>
<td>(boolean)</td>
<td>Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.</td>
</tr>
</tbody>
</table>

#### 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/wasscanschedule/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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The schedule ID. This element is assigned by the service and is required for a certain type of request.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The user-defined schedule name (maximum 256 characters).</td>
</tr>
<tr>
<td>owner.id</td>
<td>(integer) ID associated with the owner who created the schedule.</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>updatedDate</td>
<td>(date) The date when the schedule was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The scheduled scan type: VULNERABILITY or DISCOVERY.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) The name of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) The ID of the web application being scanned.</td>
</tr>
<tr>
<td>webApp.tags</td>
<td>Tags associated with the web application being scanned.</td>
</tr>
<tr>
<td>webApp.tags.id</td>
<td>(integer) ID of the tag applied to the web application being scanned.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>invalid</td>
<td>(boolean) Indicates the schedule is invalid. The web application to which the schedule was applied is deleted and hence the schedule is invalid.</td>
</tr>
<tr>
<td>lastScan (with operation=&quot;NONE&quot;)</td>
<td>(boolean) Indicates if the last scan was performed or not. True indicates that the last scan was performed.</td>
</tr>
<tr>
<td>lastScan.launchedDate</td>
<td>(date) Date when the last scan was launched on the web application, in UTC date/time format.</td>
</tr>
<tr>
<td>lastScan.status</td>
<td>(keyword) Scan status reported by last web application scan: SUBMITTED, RUNNING, FINISHED, TIME_LIMIT_EXCEEDED, SCAN_NOT_LAUNCHED, SCANNER_NOT_AVAILABLE, ERROR, CANCELED)</td>
</tr>
<tr>
<td>multi (Boolean)</td>
<td>(boolean) Indicates if the scheduled scan is single scan or multiple scan.</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(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).</td>
</tr>
<tr>
<td>name</td>
<td>(text) A report name (maximum 256 characters). Applies to all reports.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) ID of the tag associated with the report.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Name of the tag associated with the report.</td>
</tr>
<tr>
<td>creationDate</td>
<td>(date) The date when the report was created in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
</tbody>
</table>
**Qualys Web Application Scanning API**

**Reports**

| **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**

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

**XML response**

```xml
<?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**

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-ingenue/soap/soap_request.xml"
```
Qualys Web Application Scanning API

Reports

"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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(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).</td>
</tr>
<tr>
<td>name</td>
<td>(text) A report name (maximum 256 characters). Applies to all reports.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) ID of the tag associated with the report.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Name of the tag associated with the report.</td>
</tr>
<tr>
<td>creationDate</td>
<td>(date) The date when the report was created in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT,</td>
</tr>
</tbody>
</table>
Reports

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>
        </owner>
      </Report>
    </list>
  </data>
</ServiceResponse>
<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>
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**

```xml
<ServiceRequest>
  <filters>
    <Criteria field="tags.id" operator="EQUALS">99511</Criteria>
  </filters>
</ServiceRequest>
```

**XML response**

```xml
<?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>
      </Report>
    </list>
  </data>
</ServiceResponse>
```
<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>
  </Report>
</data>
</ServiceResponse>
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"
 xmlns: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>
</ServiceResponse>
Qualys Web Application Scanning API

Reports

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>
        </add>
      </distributionList>
    </Report>
  </data>
</ServiceRequest>
```
<EmailAddress><![CDATA[email2@abc.com]]></EmailAddress>
</add>
</distributionList>
</Report>
</data>
</ServiceRequest>

**XML response**

```xml
<?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**

```xml
<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

```
```

Note: “file.xml” contains the request POST data.
Qualys Web Application Scanning API

Reports

**Request POST data**

```xml
<ServiceRequest>
  <data>
    <Report>
      <tags>
        <set>
          <Tag>
            <id>99509</id>
          </Tag>
          <Tag>
            <id>99510</id>
          </Tag>
        </set>
      </tags>
    </Report>
  </data>
</ServiceRequest>
```

**XML response**

```xml
<?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**

```xml
<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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(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).</td>
</tr>
<tr>
<td>name</td>
<td>(text) A report name (maximum 256 characters). Applies to all reports.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) ID of the tag associated with the report.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Name of the tag associated with the report.</td>
</tr>
<tr>
<td>creationDate</td>
<td>(date) The date when the report was created in UTC</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API

Reports

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>
</ServiceResponse>

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**

```xml
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">to be deleted</Criteria>
    <Criteria field="status" operator="EQUALS">COMPLETE</Criteria>
  </filters>
</ServiceRequest>
```

**XML response**

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


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”.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(text) Name of the report.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) Type of the report, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT</td>
</tr>
<tr>
<td>format</td>
<td>(keyword) Report format, one of: WORD, HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, CSV, CSV_V2, XML, POWERPOINT</td>
</tr>
<tr>
<td>template.id</td>
<td>(integer) The template ID. This element is assigned by the system and is required for a certain type of request.</td>
</tr>
<tr>
<td>config*(1)</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Report Creation

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tags.id</td>
<td>(integer) ID of the tag associated with the web application.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Name of the tag associated with the web application.</td>
</tr>
<tr>
<td>password</td>
<td>(text) The password for a PDF encrypted report.</td>
</tr>
<tr>
<td>distributionList*</td>
<td>This element specifies the email addresses for distribution of the report.</td>
</tr>
</tbody>
</table>

Example:
```xml
<distributionList>
  <count>2</count>
  <list>
    <EmailAddress><![CDATA[1@abc.com]]></EmailAddress>
    <EmailAddress><![CDATA[2@abc.com]]></EmailAddress>
  </list>
</distributionList>
```

### 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**

```xml
<ServiceRequest>
  <data>
    <Report>
      <name><![CDATA[API Web Application Report]]></name>
      <description><![CDATA[PDF WebApp report]]></description>
      <format>PDF</format>
      <type>WAS_WEBAPP_REPORT</type>
    </Report>
  </data>
</ServiceRequest>
```
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>
Qualys Web Application Scanning API
Report Creation

<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>

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>
</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

```xml
<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>
            </target>
          </webAppReport>
        </config>
      </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**

```bash
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
<ServiceRequest>
  <data>
    <Report>
```

**XML response**

```xml
    <Report>
      <id>973056</id>
    </Report>
  </data>
</ServiceResponse>
```
Qualys Web Application Scanning API

Report Creation

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. See Reference: Report Creation for details.

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target.scans</td>
<td>(WasScan) The web applications to be scanned.</td>
</tr>
<tr>
<td>filters.searchlists</td>
<td>(SearchList) Number of search lists to report on vulnerabilities in those lists. If no search lists are selected, the report will include all findings.</td>
</tr>
<tr>
<td>filters.url</td>
<td>(text) Number of URLs of the web applications to being scanned.</td>
</tr>
<tr>
<td>filters.status</td>
<td>(ScanFindingStatus) Select status of vulnerabilities to be included in this report: New, Active, Re-opened, Fixed, Protected.</td>
</tr>
<tr>
<td>Filter/Setting</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>filters.remediation.showPatched</td>
<td>(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.</td>
</tr>
<tr>
<td>filters.remediation.ignoredReasons</td>
<td>(keyword) The reason to ignore a finding: FALSE_POSITIVE, RISK_ACCEPTED, NOT_APPLICABLE.</td>
</tr>
<tr>
<td>display.contents</td>
<td>(ScanAppReportContent) The report content: Description, Summary, Results, Individual Records, Details, AllResults, Appendix, Severity Levels.</td>
</tr>
<tr>
<td>display.graphs</td>
<td>(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.</td>
</tr>
<tr>
<td>display.groups</td>
<td>(ScanAppReportGroup) The group category to be included in the report: URL, OWASP, WASC, State, Category, QID, Group.</td>
</tr>
<tr>
<td>display.options</td>
<td>(rawLevels) (Urgent), 4 (Critical), 3 (Serious), 2 (Medium), 1 (Minimal)</td>
</tr>
<tr>
<td>filters.remediation.showIgnored</td>
<td>(boolean) Specify if you wish to include ignored or patched findings.</td>
</tr>
<tr>
<td>format</td>
<td>(keyword) Report format, one of: WORD, HTML_ZIPPED, HTML_BASE64, PDF,</td>
</tr>
</tbody>
</table>
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

xml version="1.0" encoding="UTF-8"?>
<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>
          </contents>
        </display>
      </config>
    </Report>
  </data>
</ServiceRequest>
<ScanReportContent>APPENDIX</ScanReportContent>
</contents>
<graphs>
  <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>
<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>
  <status>
    <ScanFindingStatus>NEW</ScanFindingStatus>
    <ScanFindingStatus>ACTIVE</ScanFindingStatus>
    <ScanFindingStatus>REOPENED</ScanFindingStatus>
    <ScanFindingStatus>FIXED</ScanFindingStatus>
  </status>
</filters>
</scanReport>
</config>
</Report>
</data>
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 ZIPPPED]]></name>
      <description><![CDATA[A scan report with ignored findings]]></description>
      <format>HTML ZIPPPED</format>
      <type>WAS SCAN REPORT</type>
      <config>
        <scanReport>
          <target>
            <scans>
              <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/report.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Report>
      <id>3629</id>
    </Report>
  </data>
</ServiceResponse>
```
<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>APPENDIX</ScanReportContent>
</contents>
<graphs>
<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>
<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>
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

```bash
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

```bash
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
<?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
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Report>
      <id>214159</id>
    </Report>
</ServiceResponse>
```
Qualys Web Application Scanning API

Report Creation

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. See Reference: Report Creation for details.

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target.webapps</td>
<td>(WebApp) The web applications to be scanned.</td>
</tr>
<tr>
<td>target.tags.included.option</td>
<td>(keyword: ALL or ANY) Decides which web applications should be included in the scan.</td>
</tr>
<tr>
<td></td>
<td>ALL : Only the web applications associated with all the specified tags are included in the scan.</td>
</tr>
<tr>
<td></td>
<td>ANY : Only the web applications associated with any of the specified tags included in the scan.</td>
</tr>
<tr>
<td>target.tags.included.tagList.Tag.id</td>
<td>(integer) The web applications</td>
</tr>
</tbody>
</table>
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   | (ScanAppReportContent) The report content: Description, Summary, Results, Individual Records, Details, AllResults, Appendix, Severity Levels. |
| target.tags.excluded.option | (Keyword) Value is ALL or ANY |
| 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 (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)

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>
        </included>
      </tags>
    </target>
    <display>
      <contents>
        <ScorecardReportContent>DESCRIPTION</ScorecardReportContent>
        <ScorecardReportContent>SUMMARY</ScorecardReportContent>
        <ScorecardReportContent>GRAPHS</ScorecardReportContent>
        <ScorecardReportContent>RESULTS</ScorecardReportContent>
      </contents>
      <graphs>
        <ScorecardReportGraph>VULNERABILITIES_BY_GROUP</ScorecardReportGraph>
        <ScorecardReportGraph>VULNERABILITIES_BY_OWASP</ScorecardReportGraph>
        <ScorecardReportGraph>VULNERABILITIES_BY_WASC</ScorecardReportGraph>
      </graphs>
      <groups>
        <scorecardReportGroup>GROUP</scorecardReportGroup>
        <scorecardReportGroup>OWASP</scorecardReportGroup>
        <scorecardReportGroup>WASC</scorecardReportGroup>
      </groups>
      <options>
        <rawLevels>false</rawLevels>
      </options>
    </display>
  </scorecardReport>
</config>
Report Creation

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
<?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. See Reference: Report Creation for details.

Click here for available operators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filters.scanDate</td>
<td>(DatetimeRange) Filter by scan date.</td>
</tr>
<tr>
<td>filters.url</td>
<td>(text) Filter by web app URL.</td>
</tr>
<tr>
<td>filters.ip</td>
<td>(text) Filter by IP address.</td>
</tr>
<tr>
<td>filters.os</td>
<td>(text) Filter by OS.</td>
</tr>
<tr>
<td>filters.status</td>
<td>(EntryStatus) Filter by status.</td>
</tr>
<tr>
<td>format</td>
<td>(keyword) Report format, one of: WORD, HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, CSV, CSV_V2, XML, POWERPOINT</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API

Report Creation

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) 5 (Urgent), 4 (Critical), 3 (Serious), 2 (Medium), 1 (Minimal)

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>

311
<contents>
  <CatalogReportContent>DESCRIPTION</CatalogReportContent>
  <CatalogReportContent>SUMMARY</CatalogReportContent>
  <CatalogReportContent>GRAPHS</CatalogReportContent>
  <CatalogReportContent>RESULTS</CatalogReportContent>
  <CatalogReportContent>INDIVIDUAL_RECORDS</CatalogReportContent>
</contents>

<graphs>
  <CatalogReportGraph>ENTRIES_ADDED_OVER_TIME</CatalogReportGraph>
  <CatalogReportGraph>ENTRIES_BY_STATUS</CatalogReportGraph>
</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"?>
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
<?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
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Report>
      <id>5629</id>
    </Report>
  </data>
</ServiceResponse>
```
<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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(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).</td>
</tr>
<tr>
<td>name</td>
<td>(text) A report name (maximum 256 characters). Applies to all reports.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT, DATALIST_REPORT.</td>
</tr>
</tbody>
</table>

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**

```bash
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**

```xml
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="EQUALS">1234</Criteria>
  </filters>
</ServiceRequest>
```

**XML response**

```xml
<?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**

```xml
<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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(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).</td>
</tr>
<tr>
<td>name</td>
<td>(text) A report name (maximum 256 characters). Applies to all reports.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT, DATALIST_REPORT.</td>
</tr>
</tbody>
</table>

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>
    </ReportTemplate>
  </data>
</ServiceResponse>
```
```xml
<remediation>
  <showPatched>SHOW_BOTH</showPatched>
  <showIgnored>SHOW_NONE</showIgnored>
  <ignoredReasons>
    <IgnoredReason>NOT_APPLICABLE</IgnoredReason>
    <IgnoredReason>FALSE_POSITIVE</IgnoredReason>
    <IgnoredReason>RISK_ACCEPTED</IgnoredReason>
  </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.comqps/rest/3.0/get/was/reporttemplate/876048"

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>

321
<![[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</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</WebAppReportGroup>
      </groups>
    </display>
  </webAppReportTemplate>
</config>
Qualys Web Application Scanning API

Report Creation

<XSD>
<platform_API_server>/qps/xsd/3.0/was/report.xsd
</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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(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).</td>
</tr>
<tr>
<td>name</td>
<td>(text) A report name (maximum 256 characters). Applies to all reports.</td>
</tr>
<tr>
<td>description</td>
<td>(text) A description of the report.</td>
</tr>
<tr>
<td>owner*</td>
<td>This element is assigned by the service and may be specified for an update request only.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>&lt;owner&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;id&gt;123056&lt;/id&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;username&gt;username&lt;/username&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;firstName&gt;&lt;![CDATA[Johns]]&gt;&lt;/firstName&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;lastName&gt;&lt;![CDATA[Smith]]&gt;&lt;/lastName&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/owner&gt;</td>
</tr>
<tr>
<td>type</td>
<td>(text) The report type, one of: WAS_SCAN_REPORT, WAS_WEBAPP_REPORT, WAS_SCORECARD_REPORT, WAS_CATALOG_REPORT, DATALIST_REPORT</td>
</tr>
<tr>
<td>format</td>
<td>(text) The format of the report, one of: HTML_ZIPPED, HTML_BASE64, PDF, PDF_ENCRYPTED, POWERPOINT, CSV, CSV_V2, XML, WORD</td>
</tr>
</tbody>
</table>
| tags*     | This element identifies the tags associated with the
password (text) The password for a PDF encrypted report.

distributionList* This element specifies the email addresses for distribution of the report.

Example:
<distributionList>
  <count>2</count>
  <list>
    <EmailAddress><![CDATA[1@abc.com]]></EmailAddress>
    <EmailAddress><![CDATA[2@abc.com]]></EmailAddress>
  </list>
</distributionList>

cfg* The configuration options for report creation.

Example:
<config>
  <webAppReport>
    <target>
      <tags>
        <Tag>
          <id>102609</id>
        </Tag>
      </tags>
    </target>
  <webapps>
Qualys Web Application Scanning API

Report Creation

```xml
<WebApp>
  <id>324538</id>
</WebApp>
</webapps>
</target>
```

<table>
<thead>
<tr>
<th><strong>status</strong></th>
<th>(keyword) The status of the report: RUNNING, ERROR or COMPLETE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>creationDate</strong></td>
<td>(date) The date when the report was created in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
<tr>
<td><strong>lastDownloadDate</strong></td>
<td>(date) The date when the report was last downloaded in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
<tr>
<td><strong>downloadCount</strong></td>
<td>(integer) The number of times the report has been downloaded.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The ID of the option profile.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The name given to the option profile.</td>
</tr>
<tr>
<td>tags</td>
<td>Filter by tags applied.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) ID of the tag assigned to option profile.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Tag name assigned to option profile.</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the option profile was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>updatedDate</td>
<td>(date) The date when the option profile was updated in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>usedByWebApps</td>
<td>(boolean) Web applications used/not used by the option profile.</td>
</tr>
<tr>
<td>usedBySchedules</td>
<td>(boolean) Scan schedules used/not used by the option profile.</td>
</tr>
<tr>
<td>owner.id</td>
<td>(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the option profile.</td>
</tr>
<tr>
<td>owner.name</td>
<td>(text) Full name of the user who created the option profile.</td>
</tr>
<tr>
<td>owner.username</td>
<td>(text) Username of the owner who created the option profile. (like acme_ab3).</td>
</tr>
</tbody>
</table>

**Sample - Count - no criteria (GET)**

**API request**

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

**XML response**

```xml
<?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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The ID of the option profile.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The name given to the option profile.</td>
</tr>
<tr>
<td>tags</td>
<td>Filter by tags applied.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) ID of the tag assigned to option profile.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Tag name assigned to option profile.</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the option profile was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>updatedDate</td>
<td>(date) The date when the option profile was updated in WAS, in UTC date/time format.</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API

Option Profiles

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usedByWebApps</td>
<td>(boolean) Web applications used/not used by the option profile.</td>
</tr>
<tr>
<td>usedBySchedules</td>
<td>(boolean) Scan schedules used/not used by the option profile.</td>
</tr>
<tr>
<td>owner.id</td>
<td>(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the option profile.</td>
</tr>
<tr>
<td>owner.name</td>
<td>(text) Full name of the user who created the option profile.</td>
</tr>
<tr>
<td>owner.username</td>
<td>(text) Username of the owner who created the option profile. (like acme_ab3).</td>
</tr>
</tbody>
</table>

Sample - Search - criteria (POST)

**API request**

```bash
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**

```xml
<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
<?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>acme_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

Sample - Get details (GET)

API request

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

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>
      </owner>
    </OptionProfile>
  </data>
</ServiceResponse>
```
<username>acme_ww</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>acme_ww</username>
      </author>
    </Comment>
    <Comment>
      <contents><![CDATA[some more comments]]></contents>
      <author>
        <id>200639085669</id>
        <username>acme_ww</username>
      </author>
    </Comment>
  </list>
</comments>
<sensitiveContent>
  <creditCardNumber>false</creditCardNumber>
  <socialSecurityNumber>false</socialSecurityNumber>
</sensitiveContent>
<createdDate>2017-09-08T22:03:01Z</createdDate>
<createdBy>
  <id>8792415669</id>
</createdBy>
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/467333"

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> 
   </owner> 
  </OptionProfile> 
 </data> 
</ServiceResponse>
<username>acme_as2</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>acme_as2</username>
  <firstName>
    <![CDATA[Ari]]>
  </firstName>
  <lastName>
    <![CDATA[Smith]]>
  </lastName>
</createdBy>
Sample - Get details on option profile with SmartScan enabled (GET)

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
<?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>
            </owner>
        </OptionProfile>
    </data>
</ServiceResponse>
```
<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>
</comments>

<list>
    <Comment>
        <contents>
            <![CDATA[User Comment]]>
        </contents>
        <createdDate>2017-11-18T15:59:55Z</createdDate>
    </Comment>
</list>

</comments>
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/1608560"
```

**XML response**

```xml
<?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>
```
<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 - custom scan intensity (POST)
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**

```xml
<ServiceRequest>
  <data>
    <OptionProfile>
      <name><![CDATA[My Option Profile - with defaults]]></name>
    </OptionProfile>
  </data>
</ServiceRequest>
```

**XML response**

```xml
<?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>879241569</id>
        <username>acme_as</username>
        <firstName><![CDATA[Alex]]></firstName>
        <lastName><![CDATA[Smith]]></lastName>
    ```
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>
    </OptionProfile>
  </data>
</ServiceRequest>
```
<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>acme_cg</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>
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>
```


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>acme_ak1</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>
        </OptionProfile>
    </data>
</ServiceResponse>
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/options.xml" < 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/options.xsd">
```

349
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 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>
            <ignoreBinaryFiles>true</ignoreBinaryFiles>
        </OptionProfile>
    </data>
</ServiceResponse>
<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>
</敏感内容>
<createdDate>2017-11-18T15:59:49Z</createdDate>
</OptionProfile>
</data>
Sample - Create - associate pre-defined detection category

Create a new option profile and associate pre-defined detection categories with Option Profile.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>detectionCategory={Keyword}</td>
<td>We now support the following new detection categories in your option profile:</td>
</tr>
<tr>
<td></td>
<td>---CMS identification (type, version, and plugins)</td>
</tr>
<tr>
<td></td>
<td>---Apache vulnerabilities (Struts &amp; other)</td>
</tr>
<tr>
<td></td>
<td>--- XSS</td>
</tr>
<tr>
<td></td>
<td>--- XSS, in request header</td>
</tr>
<tr>
<td></td>
<td>--- SQL Injection</td>
</tr>
<tr>
<td></td>
<td>--- SQL Injection, in request header</td>
</tr>
<tr>
<td></td>
<td>--- Clickjacking</td>
</tr>
<tr>
<td></td>
<td>--- Authentication &amp; Session Management</td>
</tr>
<tr>
<td></td>
<td>--- Information Disclosure</td>
</tr>
<tr>
<td></td>
<td>--- Path-Related vulnerabilities</td>
</tr>
<tr>
<td></td>
<td>--- Cross-Site Request Forgery</td>
</tr>
<tr>
<td></td>
<td>--- Open Redirect</td>
</tr>
<tr>
<td></td>
<td>--- XML External Entity (XXE) vulnerabilities</td>
</tr>
</tbody>
</table>
---Miscellaneous

---Denial of Service

---Flash-Related vulnerabilities

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**

```xml
<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
<?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">
```

355
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>Sample Option Profile With XSS</name>
      <detection>
        <xssPowerMode>true</xssPowerMode>
      </detection>
    </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>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The ID of the option profile.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The name given to the option profile.</td>
</tr>
<tr>
<td>tags</td>
<td>Filter by tags applied.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) ID of the tag assigned to option profile.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Tag name assigned to option profile.</td>
</tr>
<tr>
<td>owner.id</td>
<td>(Long with operator: EQUALS, IN, NOT EQUALS,</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API
Option Profiles

GREATER or LESSER) ID of the owner who created the option profile.

<table>
<thead>
<tr>
<th>owner.name</th>
<th>(text) Full name of the user who created the option profile.</th>
</tr>
</thead>
<tbody>
<tr>
<td>owner.username</td>
<td>(text) Username of the owner who created the option profile. (like acme_ab3).</td>
</tr>
<tr>
<td>isDefault</td>
<td>Default option profile for the subscription</td>
</tr>
<tr>
<td>formSubmission</td>
<td>(keyword) Type of form: None, Post, Get, POST&amp; GET</td>
</tr>
<tr>
<td>maxCrawlRequests</td>
<td>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</td>
</tr>
<tr>
<td>userAgent</td>
<td>Stores the browser and OS details.</td>
</tr>
<tr>
<td>parameterSet</td>
<td>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.</td>
</tr>
<tr>
<td>ignoreBinaryFiles</td>
<td>If you choose these option files with extension zip, pdf, doc are not scanned.</td>
</tr>
<tr>
<td>performance</td>
<td>(keyword) Scan Intensity: LOWEST, LOW, MEDIUM, HIGH, MAXIMUM.</td>
</tr>
<tr>
<td>customPerformance*</td>
<td>Configure the custom intensity level for web application scans.</td>
</tr>
</tbody>
</table>

Example:

```xml
<customPerformance>
  <numOfHttpThreads>10</numOfHttpThreads>
  <delayBetweenRequests>5</delayBetweenRequests>
</customPerformance>
```
### Qualys Web Application Scanning API

**Option Profiles**

Note: performance and customPerformance are mutually exclusive parameters and cannot be used together. You can use only either of them for an option profile.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numOfHttpThreads</td>
<td>(integer) Number of threads to be used to scan each host. The valid range is from 1 to 10.</td>
</tr>
<tr>
<td>delayBetweenRequests</td>
<td>(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.</td>
</tr>
<tr>
<td>bruteforceOption</td>
<td>The level of brute forcing you prefer with options ranging from “Minimal” to “Exhaustive”.</td>
</tr>
<tr>
<td>bruteforceList</td>
<td>(keyword: User List/SYSTEM LIST)</td>
</tr>
<tr>
<td></td>
<td>System list: we’ll attempt to guess the password for each detected login ID.</td>
</tr>
<tr>
<td></td>
<td>User list: to select a bruteforce list defined in your account.</td>
</tr>
<tr>
<td>numberOfAttempts</td>
<td>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.</td>
</tr>
<tr>
<td>detection</td>
<td>(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</td>
</tr>
<tr>
<td>sensitiveContent</td>
<td>Credit Card Numbers, Social Security Numbers (US), Custom Contents.</td>
</tr>
<tr>
<td>comments</td>
<td>User-defined comments.</td>
</tr>
</tbody>
</table>

### 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)

Sample - Update - minimum criteria (POST)

Change the option profile name to “Update Option Profile - title” for option profile ID 832265669.

**API request**


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

**Request POST data**

```xml
<ServiceRequest>
  <data>
    <OptionProfile>
      <name><![CDATA[Update Option Profile - title]]></name>
    </OptionProfile>
  </data>
</ServiceRequest>
```

**XML response**

```xml
<?xml version="1.0" encoding="UTF-8"?>
```
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/832275669" < 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>
  </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>
</ServiceResponse>
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/123456" < file.xml
Note: “file.xml” contains the request POST data.

**Request POST data**

```xml
<ServiceRequest>
  <data>
    <OptionProfile>
      <owner><id>123456</id></owner>
    </OptionProfile>
  </data>
</ServiceRequest>
```

**XML response**

```xml
<?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://qualsapi.qualys.com/qps/rest/3.0/update/was/optionprofile/452933" < 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://qualsapi.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://qualsapi.qualys.com/qps/rest/3.0/update/was/optionprofile/176683" < file.xml
Note: “file.xml” contains the request POST data.
Request POST data

```xml
<ServiceRequest>
  <data>
    <OptionProfile>
      <name>
        <![CDATA[My Option Profile - with action URI]]>
      </name>
      <includeActionUriInFormId>false</includeActionUriInFormId>
    </OptionProfile>
  </data>
</ServiceRequest>
```

XML response

```xml
<?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

```bash
```

Note: “file.xml” contains the request POST data.
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/1608560" < 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>
```

**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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>(text) The name given to the option profile.</td>
</tr>
<tr>
<td>owner</td>
<td>(text) Username of the owner who created the option profile. (like acme_ab3).</td>
</tr>
<tr>
<td>tags</td>
<td>(text) Filter by tags applied to option profile.</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the option profile was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>updatedDate</td>
<td>(date) The date when the option profile was updated in WAS, in UTC date/time format.</td>
</tr>
</tbody>
</table>
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

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @-
"https://qualysapi.qualys.com/qps/rest/3.0/delete/was/optionprofile/834275669"
```

XML response

```xml
<?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

```bash
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

```xml
<ServiceRequest>
  <filters>
    <Criteria field="name" operator="CONTAINS">OP</Criteria>
    <Criteria field="updatedDate" operator="LESSER">2017-09-09</Criteria>
  </filters>
</ServiceRequest>
```

XML response

```xml
<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>
  </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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) The ID of the option profile.</td>
</tr>
<tr>
<td>name</td>
<td>(text) The name given to the option profile.</td>
</tr>
<tr>
<td>tags</td>
<td>Filter by tags applied.</td>
</tr>
<tr>
<td>tags.id</td>
<td>(integer) ID of the tag assigned to option profile.</td>
</tr>
<tr>
<td>tags.name</td>
<td>(text) Tag name assigned to option profile.</td>
</tr>
<tr>
<td>createdDate</td>
<td>(date) The date when the option profile was created in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>updatedDate</td>
<td>(date) The date when the option profile was updated in WAS, in UTC date/time format.</td>
</tr>
<tr>
<td>usedByWebApps</td>
<td>(boolean) Web applications used/not used by the option profile.</td>
</tr>
<tr>
<td>usedBySchedules</td>
<td>(boolean) Scan schedules used/not used by the option profile.</td>
</tr>
<tr>
<td>owner.id</td>
<td>(Long with operator: EQUALS, IN, NOT EQUALS, GREATER or LESSER) ID of the owner who created the option profile.</td>
</tr>
<tr>
<td>owner.name</td>
<td>(text) Full name of the user who created the option profile.</td>
</tr>
<tr>
<td>owner.username</td>
<td>(text) Username of the owner who created the option profile. (like acme_ab3).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isDefault</td>
<td>Default option profile for the subscription</td>
</tr>
<tr>
<td>formSubmission</td>
<td>(keyword) Type of form: None, Post, Get, POST&amp; GET</td>
</tr>
<tr>
<td>maxCrawlRequests</td>
<td>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</td>
</tr>
<tr>
<td>userAgent</td>
<td>Stores the browser and OS details.</td>
</tr>
<tr>
<td>parameterSet</td>
<td>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.</td>
</tr>
<tr>
<td>ignoreBinaryFiles</td>
<td>If you choose these option files with extension zip, pdf, doc are not scanned.</td>
</tr>
<tr>
<td>performance</td>
<td>(keyword) Scan Intensity: LOWEST, LOW, MEDIUM, HIGH, MAXIMUM.</td>
</tr>
<tr>
<td>customPerformance*</td>
<td>Configure the custom intensity level for web application scans.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>&lt;customPerformance&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;numOfHttpThreads&gt;10&lt;/numOfHttpThreads&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;delayBetweenRequests&gt;5&lt;/delayBetweenRequests&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/customPerformance&gt;</td>
</tr>
<tr>
<td></td>
<td>Note: performance and customPerformance are mutually exclusive parameters and cannot be used together. You can use only either of them for an option profile.</td>
</tr>
<tr>
<td>numOfHttpThreads</td>
<td>(integer) Number of threads to be used to scan each host. The valid range is from 1 to 10.</td>
</tr>
<tr>
<td>delayBetweenRequests</td>
<td>(integer) The duration of delay introduced by WAS in</td>
</tr>
</tbody>
</table>
between the scanning engine requests sent to the applications server. The valid range is from 0 to 2000 milliseconds.

<table>
<thead>
<tr>
<th>Option Profiles</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bruteforceOption</td>
<td>The level of brute forcing you prefer with options ranging from “Minimal” to “Exhaustive”.</td>
</tr>
<tr>
<td>bruteforceList</td>
<td>(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.</td>
</tr>
<tr>
<td>numberOfAttempts</td>
<td>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.</td>
</tr>
<tr>
<td>detection</td>
<td>(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</td>
</tr>
<tr>
<td>sensitiveContent</td>
<td>Credit Card Numbers, Social Security Numbers (US), Custom Contents.</td>
</tr>
<tr>
<td>comments</td>
<td>User-defined comments.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Web application ID.</td>
</tr>
<tr>
<td>qid</td>
<td>(integer) Qualys ID assigned to the detection.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Name of the detection finding.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or</td>
</tr>
<tr>
<td></td>
<td>INFORMATION_GATHERED.</td>
</tr>
<tr>
<td>url</td>
<td>(text) URL of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>webApp.tags.id</td>
<td>(date) ID of the tag associated with the web</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>webApp.tags.name</td>
<td>(text) Name of the tag associated with the web application on which the finding was detected.</td>
</tr>
<tr>
<td>status</td>
<td>(keyword) Status of the finding: NEW, ACTIVE or REOPENED</td>
</tr>
<tr>
<td>patch</td>
<td>(integer-long) Use WAF to protect against vulnerabilities by installing virtual patches.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) ID of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) Name of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>severity</td>
<td>(integer) Severity of the finding.</td>
</tr>
<tr>
<td>externalRef</td>
<td>(string) Tip - Use operator IS EMPTY for findings with empty external references.</td>
</tr>
<tr>
<td>ignoredDate</td>
<td>(date) The date on which the finding was marked to ignore.</td>
</tr>
<tr>
<td>ignoredReason</td>
<td>(keyword) The reason for which the finding is ignored: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE</td>
</tr>
<tr>
<td>group</td>
<td>(keyword) XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM</td>
</tr>
<tr>
<td>owasp.name</td>
<td>(text) Name of the OWASP vulnerability.</td>
</tr>
<tr>
<td>owasp.code</td>
<td>(integer) Code associated with the OWASP vulnerability</td>
</tr>
<tr>
<td>wasc.name</td>
<td>(text) Name of the vulnerability.</td>
</tr>
<tr>
<td>wasc.code</td>
<td>(integer) Code of the vulnerability.</td>
</tr>
</tbody>
</table>
### Findings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cwe.id</td>
<td>(integer) ID associated with CWE.</td>
</tr>
<tr>
<td>firstDetectedDate</td>
<td>(date) The date when the finding was first detected in the web application.</td>
</tr>
<tr>
<td>lastDetectedDate</td>
<td>(date) The date when the finding was last detected in the web application.</td>
</tr>
<tr>
<td>lastTestedDate</td>
<td>(date) The date when the finding was last tested in the web application.</td>
</tr>
<tr>
<td>timesDetected</td>
<td>(integer) The count indicating the number of times the finding was detected.</td>
</tr>
<tr>
<td>severity level</td>
<td>(integer) The severity associated with the finding: 1, 2, 3, 4, 5</td>
</tr>
</tbody>
</table>

### 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
<?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>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/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://qualysapi.qualys.com/qps/xsd/3.0/was/finding.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>41</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.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Web application ID.</td>
</tr>
<tr>
<td>qid</td>
<td>(integer) Qualys ID assigned to the detection.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Name of the detection finding.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.</td>
</tr>
<tr>
<td>url</td>
<td>(text) URL of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>webApp.tags.id</td>
<td>(date) ID of the tag associated with the web application on which the finding was detected.</td>
</tr>
</tbody>
</table>
### Qualys Web Application Scanning API

#### Findings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webApp.tags.name</td>
<td>(text) Name of the tag associated with the web application on which the finding was detected.</td>
</tr>
<tr>
<td>status</td>
<td>(keyword) Status of the finding: NEW, ACTIVE or REOPENED</td>
</tr>
<tr>
<td>patch</td>
<td>(integer-long) Use WAF to protect against vulnerabilities by installing virtual patches.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) ID of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) Name of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>severity</td>
<td>(integer) Severity of the finding.</td>
</tr>
<tr>
<td>externalRef</td>
<td>(string) Tip - Use operator IS EMPTY for findings with empty external references.</td>
</tr>
<tr>
<td>ignoredDate</td>
<td>(date) The date on which the finding was marked to ignore.</td>
</tr>
<tr>
<td>ignoredReason</td>
<td>(keyword) The reason for which the finding is ignored: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE</td>
</tr>
<tr>
<td>group</td>
<td>(keyword) XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM</td>
</tr>
<tr>
<td>owasp.name</td>
<td>(text) Name of the OWASP vulnerability.</td>
</tr>
<tr>
<td>owasp.code</td>
<td>(integer) Code associated with the OWASP vulnerability</td>
</tr>
<tr>
<td>wasc.name</td>
<td>(text) Name of the vulnerability.</td>
</tr>
<tr>
<td>wasc.code</td>
<td>(integer) Code of the vulnerability.</td>
</tr>
<tr>
<td>cwe.id</td>
<td>(integer) ID associated with CWE.</td>
</tr>
</tbody>
</table>
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 - 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
<?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>
    <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>
        </OWASP>
      </list>
    </owasp>
    <wasc>
      <count>1</count>
      <list>
        <WASC>
          <name><![CDATA[Application Misconfiguration]]></name>
          <url>http://projects.webappsec.org/w/page/13246914/WASC</url>
        </WASC>
      </list>
    </wasc>
  </Finding>
</data>
<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/xss/]]>
</url>
$status>ACTIVE</status>
Qualys Web Application Scanning API

Findings

Sample - Search with criteria: condensed response

API request

```bash
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

```xml
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="EQUALS">935943</Criteria>
  </filters>
</ServiceRequest>
```

XML response

```xml
<?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">
  ...
</ServiceResponse>
```
<responseCode>SUCCESS</responseCode>
<count>1</count>
<hasMoreRecords>false</hasMoreRecords>
<data>
  <Finding>
    <id>935943</id>
    <qid>150117</qid>
    <name><![CDATA[Path-Based Cross-Site Scripting (XSS)]]></name>
    <type>VULNERABILITY</type>
    <findingType>AFCO</findingType>
    <severity>5</severity>
    <url><![CDATA[http://funkytown.vuln.qa.afco.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.afco.com:80/cassium/xss/]]></url>
    </webApp>
    <isIgnored>true</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.

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>
      <qid>150117</qid>
    </Finding>
  </data>
</ServiceResponse>
<name>
  <![CDATA[Path-Based Cross-Site Scripting (XSS)]]>
</name>
<type>VULNERABILITY</type>
<findingType>AFCO</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/Top_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>
      
    </Result>
  </list>
</resultList>
<authentication>false</authentication>
<ajax>false</ajax>
<payloads>
  <list>
    <PayloadInstance>
      <payload>
        <![CDATA[@APPEND@/%22%3e%3cimg
%20src%3dq%20onerror%3dalert(9)%3e]]>
      </payload>
      <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[UmVmZXJlcjogaHR0cDovL2Z1bmt5dG93bi52dWxuLnFhLnFiYWyx5cy5jbjb20v5Y2Fzc21ib594c3MvDQpDb29raWU6IFBIUFNFU1NJRD00ODlmNTI4ZjUxNWE1MTY3MjM0OTQwNzExYTE1MWM0MDsNCg==]]></headers>
      <response><![CDATA[<html><head><title>Welcome to page page_48/</title></head><body><h1>Welcome to page page_48/</h1><img src=q onerror=alert(9)></img><img src=q onerror=alert(9)></img>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>
      <payloadResponse>
        <offset>16</offset>
        <length>62</length>
      </payloadResponse>
    </PayloadInstance>
  </list>
</payloads>
</Result>
</resultList>
<severity>5</severity>
<url>
<![CDATA[http://funkytown.vuln.qa.afco.com/cassium/traversal/page_48/%22%3e%3cimg%src%3dq%onerror%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.afco.com:80/cassium/xss/]]></url>
</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>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Web application ID.</td>
</tr>
<tr>
<td>qid</td>
<td>(integer) Qualys ID assigned to the detection.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Name of the detection finding.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.</td>
</tr>
<tr>
<td>url</td>
<td>(text) URL of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>webApp.tags.id</td>
<td>(date) ID of the tag associated with the web application on which the finding was detected.</td>
</tr>
</tbody>
</table>
### Findings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webApp.tags.name</td>
<td>(text) Name of the tag associated with the web application on which the finding was detected.</td>
</tr>
<tr>
<td>status</td>
<td>(keyword) Status of the finding: NEW, ACTIVE or REOPENED</td>
</tr>
<tr>
<td>patch</td>
<td>(integer-long) Use WAF to protect against vulnerabilities by installing virtual patches.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) ID of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) Name of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>severity</td>
<td>(integer) Severity of the finding.</td>
</tr>
<tr>
<td>externalRef</td>
<td>(string) Tip - Use operator IS EMPTY for findings with empty external references.</td>
</tr>
<tr>
<td>ignoredDate</td>
<td>(date) The date on which the finding was marked to ignore.</td>
</tr>
<tr>
<td>ignoredReason</td>
<td>(keyword) The reason for which the finding is ignored: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE</td>
</tr>
<tr>
<td>group</td>
<td>(keyword) XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM</td>
</tr>
<tr>
<td>reactivateDate</td>
<td>(date) Specify the date after which the ignored finding should be re-activated. The date/time is specified in YYYY-MM-DD format.</td>
</tr>
<tr>
<td>reactivateIn</td>
<td>(integer) Specify the number of days after which the ignored finding should be reactivated.</td>
</tr>
</tbody>
</table>

Note: reactivateDate and reactivateIn are mutually exclusive parameters and cannot be used together. You can use only either of them for a finding.
Qualys Web Application Scanning API

Findings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>owasp.name</td>
<td>(text) Name of the OWASP vulnerability.</td>
</tr>
<tr>
<td>owasp.code</td>
<td>(integer) Code associated with the OWASP vulnerability</td>
</tr>
<tr>
<td>wasc.name</td>
<td>(text) Name of the vulnerability.</td>
</tr>
<tr>
<td>wasc.code</td>
<td>(integer) Code of the vulnerability.</td>
</tr>
<tr>
<td>cwe.id</td>
<td>(integer) ID associated with CWE.</td>
</tr>
<tr>
<td>firstDetectedDate</td>
<td>(date) The date when the finding was first detected in the web application,</td>
</tr>
<tr>
<td>lastDetectedDate</td>
<td>(date) The date when the finding was last detected in the web application.</td>
</tr>
<tr>
<td>lastTestedDate</td>
<td>(date) The date when the finding was last tested in the web application.</td>
</tr>
<tr>
<td>timesDetected</td>
<td>(integer) The count indicating the number of times the finding was detected.</td>
</tr>
<tr>
<td>severity level</td>
<td>(integer) The severity associated with the finding: 1, 2, 3, 4, 5</td>
</tr>
</tbody>
</table>

Sample - Ignore a specific finding

API request

curl -n -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/3.0/ignore/was/finding/1645195669"

Request POST data

```xml
<ServiceRequest>
  <data>
    <Finding>
      <id>1645195669</id>
    </Finding>
  </data>
</ServiceRequest>
```
XML response

```xml
<?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>
    </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

```xml
<ServiceRequest>
  <data>
    <Finding>
      <id>927823</id>
      <ignoredReason>FALSE_POSITIVE</ignoredReason>
      <ignoredComment>test</ignoredComment>
      <reactivateDate>2018-11-14</reactivateDate>
    </Finding>
  </data>
</ServiceRequest>
```
XML response

```xml
<?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>
    </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>927823</id>
    </Finding>
  </data>
</ServiceResponse>
```
Qualys Web Application Scanning API

Findings

XSD

<xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/3.0/was/finding.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Finding>
      <id>927913</id>
    </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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Web application ID.</td>
</tr>
<tr>
<td>qid</td>
<td>(integer) Qualys ID assigned to the detection.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Name of the detection finding.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.</td>
</tr>
<tr>
<td>url</td>
<td>(text) URL of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>webApp.tags.id</td>
<td>(date) ID of the tag associated with the web application on which the finding was detected.</td>
</tr>
</tbody>
</table>
## Qualys Web Application Scanning API

### Findings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>webApp.tags.name</code></td>
<td>(text) Name of the tag associated with the web application on which the finding was detected.</td>
</tr>
<tr>
<td><code>status</code></td>
<td>(keyword) Status of the finding: NEW, ACTIVE or REOPENED</td>
</tr>
<tr>
<td><code>patch</code></td>
<td>(integer-long) Use WAF to protect against vulnerabilities by installing virtual patches.</td>
</tr>
<tr>
<td><code>webApp.id</code></td>
<td>(integer) ID of the web application on which the finding was detected.</td>
</tr>
<tr>
<td><code>webApp.name</code></td>
<td>(text) Name of the web application on which the finding was detected.</td>
</tr>
<tr>
<td><code>severity</code></td>
<td>(integer) Severity of the finding.</td>
</tr>
<tr>
<td><code>externalRef</code></td>
<td>(string) Tip - Use operator IS EMPTY for findings with empty external references.</td>
</tr>
<tr>
<td><code>ignoredDate</code></td>
<td>(date) The date on which the finding was marked to ignore.</td>
</tr>
<tr>
<td><code>ignoredReason</code></td>
<td>(keyword) The reason for which the finding is ignored: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE</td>
</tr>
<tr>
<td><code>group</code></td>
<td>(keyword) XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM</td>
</tr>
<tr>
<td><code>owasp.name</code></td>
<td>(text) Name of the OWASP vulnerability.</td>
</tr>
<tr>
<td><code>owasp.code</code></td>
<td>(integer) Code associated with the OWASP vulnerability</td>
</tr>
<tr>
<td><code>wasc.name</code></td>
<td>(text) Name of the vulnerability.</td>
</tr>
<tr>
<td><code>wasc.code</code></td>
<td>(integer) Code of the vulnerability.</td>
</tr>
<tr>
<td><code>cwe.id</code></td>
<td>(integer) ID associated with CWE.</td>
</tr>
</tbody>
</table>
**Qualys Web Application Scanning API**

**Findings**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>firstDetectedDate</td>
<td>(date) The date when the finding was first detected in the web application.</td>
</tr>
<tr>
<td>lastDetectedDate</td>
<td>(date) The date when the finding was last detected in the web application.</td>
</tr>
<tr>
<td>lastTestedDate</td>
<td>(date) The date when the finding was last tested in the web application.</td>
</tr>
<tr>
<td>timesDetected</td>
<td>(integer) The count indicating the number of times the finding was detected.</td>
</tr>
<tr>
<td>severity level</td>
<td>(integer) The severity associated with the finding: 1, 2, 3, 4, 5</td>
</tr>
</tbody>
</table>

**Sample - Activate all ignored findings**

**API request**

```bash
curl -n -u "USERNAME:PASSWORD" "qualysapi.qualys.com/qps/rest/3.0/activate/was/finding"
```

**XML response**

```xml
<?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>
    </Finding>
    <Finding>
      <id>1613255669</id>
    </Finding>
    <Finding>
      <id>1645195669</id>
    </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>
        </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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Finding ID.</td>
</tr>
<tr>
<td>new severity level</td>
<td>(integer) {1,2,3,4,5}</td>
</tr>
<tr>
<td>comments</td>
<td>(text) User comments.</td>
</tr>
</tbody>
</table>

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

```xml
<ServiceRequest>
  <data>
    <Finding>
      <id>647</id>
      <severityComment>Test comment API</severityComment>
      <severity>2</severity>
    </Finding>
  </data>
</ServiceRequest>
```

XML response

```xml
<?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>
    </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>
        </Finding>
        <Finding>
            <id>183</id>
        </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, WebApplg, or WebAppSensitiveContent).

Click here for available operators

**Sample - Restore severity level**

**API request**

```bash
curl -n -u "USERNAME:PASSWORD"
"qualysapi.qualys.com/qps/rest/3.0/restoreSeverity/was/finding"
```

**Request POST data**

```xml
<ServiceRequest>
  <data>
    <Finding>
      <id>6034</id>
    </Finding>
  </data>
</ServiceRequest>
```
### XML response

```xml
<?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>
    </Finding>
  </data>
</ServiceResponse>
```

### Sample - Restore for multiple findings

**API request**

```bash
curl -n -u "USERNAME:PASSWORD" 
"qualysapi.qualys.com/qps/rest/3.0/restoreSeverity/was/finding"
```

**Request POST data**

```xml
<ServiceRequest>
  <filters>
    <Criteria field="id" operator="IN">183, 645</Criteria>
  </filters>
</ServiceRequest>
```

**XML response**

```xml
<?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>183</id>
    </Finding>
    <Finding>
      <id>645</id>
    </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, WebAppIgl, or WebAppSensitiveContent).

Click here for available operators

Sample - Retest Finding using XML Request

**API request**

```bash
curl -n -u "USERNAME:PASSWORD" "qualysapi.qualys.com/qps/rest/3.0/retest/was/finding"
```

**Request POST data**

```xml
<ServiceRequest>
  <data>
    <Finding>
      <id>1728792</id>
    </Finding>
  </data>
</ServiceRequest>
```
Qualys Web Application Scanning API

Findings

XML response

```xml
<?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>
  </data>
</ServiceResponse>
```

Sample - Using Finding ID

API request

```bash
curl -n -u "USERNAME:PASSWORD"
"qualysapi.qualys.com/qps/rest/3.0/retest/was/finding/1728792"
```

XML response

```xml
<?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>
  </data>
</ServiceResponse>
```

XSD

```xml
<platform_API_server>/qps/xsd/3.0/was/finding.xsd
```
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:

```xml
<FINDING>
  <PAYLOAD><![CDATA[uid=%00%3Cscript%3E_q%3Drandom(X157105156Y1Z)%3C%2Script %3E]]></PAYLOAD>
  <RESULT base64="true"><![CDATA[C19mZWVkKCgKCuCI=]]></RESULT>
</FINDING>
```

If the “base64=true attribute” is not set, the value will be in plain text. For example:

```xml
<FINDING>
  <PAYLOAD><![CDATA[uid=%00%3Cscript%3E_q%3Drandom(X157105156Y1Z)%3C%2Script %3E]]></PAYLOAD>
  <RESULT><![CDATA[_feed("]]]]></RESULT>
</FINDING>
```

Which WAS reports show findings?

- WAS v3 Scan Results
Qualys Web Application Scanning API

Findings

- 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

```xml
<WasScanVuln>
  <qid>150001</qid>
  <title><![CDATA[Reflected Cross-Site Scripting (XSS) Vulnerabilities]]></title>
  <uri><![CDATA[http://myuri.apps.com/613460625329/feed.gtl?uid=%22%3E
%3Cqss%20a%3DX157105156Y1Z%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]]>
```
Qualys Web Application Scanning API

Findings

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[RGMvYXVsdCBmb3JtIGFjdGlvbiBkb2VzIG5vdCBzdWJtaXQgdmlhdGhotb3Z0Q0NTTDoGaHRoDovLd2vb2dsZS1ncnV5ZXJ1LmFwchHNwb3QuY29tLzYxMzQ2MDYyNTM0L2dpbgo=]]></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

```xml
<VULNERABILITY>
  <ID>5943</ID>
  <QID>150001</QID>
  <URL><![CDATA[http://myuri.apps.com/app/xss/0/1/0/xss.php?s='%20onEvent%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[bGQiJmd0oYZsdDsmbHQ7L3NwYW4mZ3Q7ID0mZ3Q7ICZsdDtzcFuIGNsYXNzPSJib2xkiIZndDsmYW1w02x00yZsdDsvc3Bhb1ZndDsmbHQ7YnImZ3Q7CiczsdsvZG12Jmd0owmbHQ7L2RpiZndDsmKJmx02JyJmd0owmbHQ7ZG12IGNsYXNzPSJwYXlsb2FkcyImZ3Q7Ck91dHB1dCBmc9tIHJlciXVlc3QgJmx003NwYW4gY2xhc3M9ImJvbgGQiJmd0oy9jYXNzaVtL3hzcy5wAHAdMFAwFudD0wJmFtcDtxcz0xJmFtcDtmPTAmYW1wO3M9jUyMG9uRXZ1bnQ1M2RYMTQ2NDcwMTgwTFAJItIwJmx00y9zcGFuJmd0owmbHQ7YNiZ3Q7CiZsdDthIGhyZmY9J1wnIG9uRXZ1bnQ9wDE0NjQ3MDE4MFkxWiAnJmd03Nh6XB]]></CONTENTS>
    </PAYLOAD>
  </PAYLOADS>
</VULNERABILITY>
```
Qualys Web Application Scanning API

Findings

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[I3RhYmxlClk1QX2FkJ1c3MgSG9zdF9uYW1lCgoxMC4xMC4yNi43NyBmdW5reXRvd24udnVsbib5xYS5xdWFseXMuY29tCg==]]></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[VGltaW51dCBYb25vdCBhIGluOmVQaGFiZUNyXydsCkNyXydsIGNvbXBsZXRlZCB3aXRoIFdlYktpdC4K]]></DATA>
  </INFORMATION_GATHERED>
</INFORMATION_GATHERED_LIST>
### 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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>(integer) Web application ID.</td>
</tr>
<tr>
<td>qid</td>
<td>(integer) Qualys ID assigned to the detection.</td>
</tr>
<tr>
<td>name</td>
<td>(text) Name of the detection finding.</td>
</tr>
<tr>
<td>type</td>
<td>(keyword) Type of the finding: VULNERABILITY, SENSITIVE_CONTENT, or INFORMATION_GATHERED.</td>
</tr>
<tr>
<td>url</td>
<td>(text) URL of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>webApp.tags.id</td>
<td>(integer) ID of the tag associated with the web application on which the finding was detected.</td>
</tr>
<tr>
<td>webApp.tags.name</td>
<td>(text) Name of the tag associated with the web application on which the finding was detected.</td>
</tr>
<tr>
<td>status</td>
<td>(keyword) Status of the finding: NEW, ACTIVE or REOPENED</td>
</tr>
<tr>
<td>patch</td>
<td>(integer-long) Use WAF to protect against vulnerabilities by installing virtual patches.</td>
</tr>
<tr>
<td>webApp.id</td>
<td>(integer) ID of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>webApp.name</td>
<td>(text) Name of the web application on which the finding was detected.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>severity</td>
<td>(integer) Severity of the finding.</td>
</tr>
<tr>
<td>externalRef</td>
<td>(string) Tip - Use operator IS EMPTY for findings with empty external references.</td>
</tr>
<tr>
<td>ignoredDate</td>
<td>(date) The date on which the finding was marked to ignore.</td>
</tr>
<tr>
<td>ignoredReason</td>
<td>(keyword) The reason for which the finding is ignored: FALSE_POSITIVE, RISK_ACCEPTED or NOT_APPLICABLE</td>
</tr>
<tr>
<td>group</td>
<td>(keyword) XSS, SQL, INFO, PATH, CC, SSN_US or CUSTOM</td>
</tr>
<tr>
<td>owasp.name</td>
<td>(text) Name of the OWASP vulnerability.</td>
</tr>
<tr>
<td>owasp.code</td>
<td>(integer) Code associated with the OWASP vulnerability</td>
</tr>
<tr>
<td>wasc.name</td>
<td>(text) Name of the vulnerability.</td>
</tr>
<tr>
<td>wasc.code</td>
<td>(integer) Code of the vulnerability.</td>
</tr>
<tr>
<td>cwe.id</td>
<td>(integer) ID associated with CWE.</td>
</tr>
<tr>
<td>firstDetectedDate</td>
<td>(date) The date when the finding was first detected in the web application.</td>
</tr>
<tr>
<td>lastDetectedDate</td>
<td>(date) The date when the finding was last detected in the web application.</td>
</tr>
<tr>
<td>lastTestedDate</td>
<td>(date) The date when the finding was last tested in the web application.</td>
</tr>
<tr>
<td>timesDetected</td>
<td>(integer) The count indicating the number of times the finding was detected.</td>
</tr>
<tr>
<td>severity level</td>
<td>(integer) The severity associated with the finding:1,2,3,4,5</td>
</tr>
</tbody>
</table>
# Burp

## Import Burp Scan Report

/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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webApplId</td>
<td>(integer) The web application ID. This element is assigned by the service and required for an update request.</td>
</tr>
<tr>
<td>purgeResults</td>
<td>(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: <code>&lt;purgeResults&gt;false&lt;/purgeResults&gt;</code></td>
</tr>
<tr>
<td>closeUnreportedIssues</td>
<td>(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.</td>
</tr>
</tbody>
</table>
Qualys Web Application Scanning API

Burp

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>8223303</webAppId>
    <purgeResults>false</purgeResults>
    <closeUnreportedIssues>false</closeUnreportedIssues>
    <fileName>testBurpReportImport</fileName>
    <burpXml>
      <!DOCTYPE issues [ <!ELEMENT issues (issue*)>]
      <!ATTLIST issues burpVersion CDATA "">  
      <!ATTLIST issues exportTime CDATA ">
      <!ELEMENT issue (serialNumber, requestresponse*)>
      <!ELEMENT serialNumber (#PCDATA)>
      <!ELEMENT requestresponse (request?, response?, responseRedirected?)>
      <!ELEMENT request (#PCDATA)>
      <!ATTLIST request base64 (true|false) "false">
      <!ELEMENT response (#PCDATA)>
      <!ATTLIST response base64 (true|false) "false">
      <!ELEMENT responseRedirected (#PCDATA)>]
    </burpXml>
  </data>
</ServiceRequest>
<issues burpVersion="1.5.08" exportTime="Wed May 15 12:14:53 CDT 2013">
  <issue>
    <serialNumber>7744774629008832512</serialNumber>
    <type>8389120</type>
    <name>HTML does not specify charset</name>
    <host ip="54.243.54.81">http://54.243.54.81:8080</host>
    <path>
      <![CDATA[/bodgeit/about.jsp]]>
    </path>
    <location>
      <![CDATA[/bodgeit/about.jsp]]>
    </location>
    <severity>Information</severity>
    <confidence>Certain</confidence>
    <issueBackground>
      <![CDATA[If a web response states that it contains HTML content but does not specify a character set, then the browser may analyze the HTML and attempt to determine which character set it appears to be using. Even if the majority of the HTML actually employs a standard character set such as UTF-8, the presence of non-standard characters anywhere in the response may cause the browser to interpret the content using a different character set. This can have unexpected results, and can lead to cross-site scripting vulnerabilities in which non-standard encodings like UTF-7 can be used to bypass the application's defensive filters.]]>
    </issueBackground>
    <remediationBackground>
      <![CDATA[For every response containing HTML content, the application should include within the Content-type header a directive specifying a standard recognized character set, for example <b>charset=ISO-8859-1</b>]]>
    </remediationBackground>
  </issue>
</issues>
Qualys Web Application Scanning API

Burp

```xml
<request base64="true">
<![CDATA[SFBUUCE8xLjEgMjAwMDNEMUFenzy4NzhGRDc3RjU4MDdEOg5D3Q0Nzg=]]>
</request>
</response>
```
Qualys Web Application Scanning API

Burp

XML response

```xml
<?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>226801</id>
      <webApp>
        <id>8223303</id>
        <name><![CDATA[AFCO bank]]></name>
        <url><![CDATA[http://bank.vuln.afco.com]]></url>
      </webApp>
      <issuesCount>1</issuesCount>
      <issues burpVersion="1.5.08" exportTime="Wed May 15 17:14:53 UTC 2013">
        <issue>
          <id>202001</id>
          <serialNumber>7744774629008832512</serialNumber>
        </issue>
      </issues>
    </Burp>
  </data>
</ServiceResponse>
```
XSD

<platform API server>/qps/xsd/3.0/was/burp.xsd
## Error Messages

### Sample Messages: Elements

Sample messages for element errors are shown below

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Validation</td>
<td></td>
</tr>
<tr>
<td>url: Invalid URL format (&lt;value&gt;).</td>
<td>URL format must be as follows:</td>
</tr>
<tr>
<td></td>
<td>http://&lt;baseUrl&gt;/rest/3.0/?parameters</td>
</tr>
<tr>
<td>&lt;scope&gt;: Invalid value (&lt;value&gt;).</td>
<td>Element must be set to one of these values: ALL, LIMIT, SUBDOMAIN or DOMAINS.</td>
</tr>
<tr>
<td>domains: Element is required when scope is set to: DOMAINS.</td>
<td>Specify the domains to include in the web application scope in the “domains” element.</td>
</tr>
<tr>
<td>subDomain: Element is required when scope is set to: SUBDOMAIN.</td>
<td>Specify the subdomains to include in the web application scope in the “subDomain” element.</td>
</tr>
<tr>
<td>subDomain: Invalid domain name format (&lt;value&gt;).</td>
<td>Use following format in the “subDomain” element: .my.domain.suffix (must start with a dot)</td>
</tr>
<tr>
<td>useRobots: Invalid value (&lt;value&gt;).</td>
<td>Element “userRobots” must be set to one of these values: IGNORE, ADD_PATHS, BLACKLIST.</td>
</tr>
<tr>
<td>Url: Element is required</td>
<td>Element “Url” is required.</td>
</tr>
</tbody>
</table>
### Error Messages

<table>
<thead>
<tr>
<th>Error Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uris.&lt;field&gt;: Invalid URL format (&lt;value&gt;).</td>
<td>For the uri.&lt;field&gt; sub element, specify a URL like <a href="http://domain.name/base/url/?parameters">http://domain.name/base/url/?parameters</a></td>
</tr>
<tr>
<td>uris.&lt;field&gt;: Length of the field must not be greater than 2048 characters.</td>
<td>For the uri.&lt;field&gt; sub element, the maximum field length is 2048 characters.</td>
</tr>
<tr>
<td>Domain: Element is required</td>
<td>The domain element must be provided.</td>
</tr>
<tr>
<td>Domain: Invalid host name format (&lt;value&gt;).</td>
<td>Use following format for value in the “Domain” element: <a href="http://www.my.domain.example">www.my.domain.example</a>.</td>
</tr>
<tr>
<td>Length of all domains cannot exceed 2048 characters.</td>
<td>The list of all domains in the web application cannot exceed 2048 characters.</td>
</tr>
<tr>
<td>Attribute.category: Element is required</td>
<td>The element Attribute.category is required.</td>
</tr>
<tr>
<td>Attribute.category: Invalid value (&lt;value&gt;).</td>
<td>Element Attribute.category must be set to one of these values: Business Function, Business Location, Business Description.</td>
</tr>
<tr>
<td>Attribute.value: Element is required</td>
<td>Provide a value for the attribute in the Attribute.value element: function, location or description.</td>
</tr>
<tr>
<td>The attribute length cannot be greater than 64 characters.</td>
<td>The value for this attribute cannot exceed 64 characters.</td>
</tr>
</tbody>
</table>
### Qualys Web Application Scanning API

#### Error Messages

<table>
<thead>
<tr>
<th>Error Description</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>The attribute length cannot be greater than 2048 characters.</td>
<td>The value for this attribute cannot exceed 2048 characters.</td>
</tr>
<tr>
<td>&lt;element&gt;: Element must not be set.</td>
<td>This element does not apply to this request.</td>
</tr>
<tr>
<td>set: Element must contain at least one child.</td>
<td>The set element requires at least one sub element.</td>
</tr>
<tr>
<td>At least one of the following elements must be set: set, add, remove.</td>
<td>This request requires at least one of these elements: set, add or remove.</td>
</tr>
<tr>
<td>headers: Length of all headers cannot exceed 2048 characters.</td>
<td>The values of all headers cannot exceed 2048 characters.</td>
</tr>
<tr>
<td>At least one of the following elements must be set: set, add, remove.</td>
<td>For an “update” request you must set at least one of these elements: set, add or remove.</td>
</tr>
<tr>
<td>UrlEntry: Element is required.</td>
<td>The element UrlEntry must be provided.</td>
</tr>
<tr>
<td>UrlEntry: Invalid URL format (value).</td>
<td>Specify a URL like <a href="http://domain.name/base/url/?parameters">http://domain.name/base/url/?parameters</a></td>
</tr>
<tr>
<td>&lt;parent&gt;: Length of all [URLs, regular expressions] cannot exceed 2048 characters.</td>
<td>The list of entries for a given type shall not exceed 2048 characters.</td>
</tr>
</tbody>
</table>
## Error Messages

<table>
<thead>
<tr>
<th>Description</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>UrlEntry: Only regular expressions are accepted for this element.</td>
<td>You must provide regular expressions for the element postDataBlackList.</td>
</tr>
<tr>
<td>tags.&lt;element&gt;: Element must not be set.</td>
<td>The tags element does not apply for this request.</td>
</tr>
<tr>
<td>tags.set: Element must contain at least one child.</td>
<td>At least one sub element must be provided for the element tag.set.</td>
</tr>
<tr>
<td>Tag.id: Element is required.</td>
<td>Provide a value for the element Tag.id</td>
</tr>
<tr>
<td>Tag.id: Invalid value (value).</td>
<td>Value must be an integer set at least to 1.</td>
</tr>
<tr>
<td>Tag: Tag specified by ID &lt;id&gt; does not exist or is not available.</td>
<td>Provide a value for the element id that corresponds to a valid tag.</td>
</tr>
</tbody>
</table>
Sample Messages: Authorization

Sample messages for errors related to authorization are shown below.

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Validation</td>
<td>You must be granted the API Access permission in your roles and scopes.</td>
</tr>
<tr>
<td>You are not authorized to access the</td>
<td></td>
</tr>
<tr>
<td>application through the API.</td>
<td></td>
</tr>
<tr>
<td>You do not have access to module</td>
<td>Please contact your account manager to have WAS enabled in your subscription.</td>
</tr>
<tr>
<td>Web Application Scanning required by this API.</td>
<td></td>
</tr>
<tr>
<td>No data shall be passed for this</td>
<td>The POST request does not specify a data element.</td>
</tr>
<tr>
<td>operation.</td>
<td></td>
</tr>
<tr>
<td>User is not authorized to perform this</td>
<td>You must be granted access to these objects in your user scope.</td>
</tr>
<tr>
<td>operation on specified object(s).</td>
<td></td>
</tr>
<tr>
<td>Operation %s does not support search</td>
<td>Do not provide search files for this operation</td>
</tr>
<tr>
<td>filters.</td>
<td></td>
</tr>
<tr>
<td>Quota of web application has been</td>
<td>Please check with your account manager to purchase new applications.</td>
</tr>
<tr>
<td>exceeded.</td>
<td></td>
</tr>
</tbody>
</table>
Sample Messages: Criteria

Sample messages for errors related to criteria are shown below.

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Validation</td>
<td></td>
</tr>
<tr>
<td>Criteria: Field is required.</td>
<td>Specify the name of the criteria to search against.</td>
</tr>
<tr>
<td>Criteria: Invalid criteria (&lt;field name&gt;).</td>
<td>Please search against one of the following criteria: %s.</td>
</tr>
<tr>
<td>Criteria: Invalid operator for criteria '&lt;field&gt;' (&lt;operator&gt;).</td>
<td>Allowed operations for this criteria are: %s.</td>
</tr>
<tr>
<td>Criteria: Value is required for criteria '&lt;field&gt;'.</td>
<td>Specify a value for a field name for search criteria.</td>
</tr>
<tr>
<td>Criteria: Invalid value format for criteria '&lt;field&gt;': &lt;value&gt;.</td>
<td>Boolean (true, false).</td>
</tr>
<tr>
<td></td>
<td>Date and Time in UTC format</td>
</tr>
<tr>
<td></td>
<td>Enumeration (allowed options separated by comma).</td>
</tr>
<tr>
<td></td>
<td>Other: Specify criteria value(s) as &lt;type&gt;.</td>
</tr>
</tbody>
</table>
Sample Messages: Report Storage Limit

Sample messages for errors related to report storage limit are shown below.

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Validation</td>
<td>Delete existing reports and try again.</td>
</tr>
<tr>
<td></td>
<td>Your [subscription</td>
</tr>
</tbody>
</table>