Preface

Using the Qualys Continuous Monitoring (CM) API, third parties can integrate the Qualys Security and Compliance solution into their own applications using an extensible XML interface. This user guide is intended for application developers who will use the Qualys CM API.

About Qualys

Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of cloud-based security and compliance solutions with over 9,300 customers in more than 100 countries, including a majority of each of the Forbes Global 100 and Fortune 100. The Qualys Cloud Platform and integrated suite of solutions help organizations simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications. Founded in 1999, Qualys has established strategic partnerships with leading managed service providers and consulting organizations including Accenture, BT, Cognizant Technology Solutions, Fujitsu, HCL Comnet, HPE, Infosys, NTT, Optiv, SecureWorks, Tata Communications, Verizon and Wipro. The company is also a founding member of the Cloud Security Alliance (CSA).

For more information, please visit www.qualys.com.

Contact Qualys Support

Qualys is committed to providing you with the most thorough support. Through online documentation, telephone help, and direct email support, Qualys ensures that your questions will be answered in the fastest time possible. We support you 7 days a week, 24 hours a day. Access support information at www.qualys.com/support/.
Preface
Welcome

Welcome to Qualys Continuous Monitoring API. Several functional suites are available to support CM.

Get Started

Introduction to the CM API Paradigm - Review important information about the CM API framework.

Base URL to the Qualys API Server - Learn the basics about making API requests. The base URL depends on the platform where your Qualys account is located.

Authentication - We’ll tell you about the method used for authentication. API requests must authenticate using Qualys credentials.

Get API Notifications

We recommend you join our Community and subscribe to our API notifications so you’ll get email notifications telling you about important upcoming API enhancements and changes.

From our Community

Join our Community
Subscribe to API Notifications (select Receive email notifications)
Introduction to the CM API Paradigm

The new Qualys CM API framework introduces numerous innovations and new functionality compared to the other Qualys API frameworks.

Request URL

The URL for making API requests respects the following structure:

https://<baseurl>/qps/rest/1.0/<operation>/<module>/<object>/<object_id>

where the components are described below.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;baseurl&gt;</td>
<td>The Qualys API server URL that you should use for API requests depends on the platform where your account is located. The base URL for Qualys US Platform 1 is: <a href="https://qualysapi.qualys.com">https://qualysapi.qualys.com</a></td>
</tr>
<tr>
<td>&lt;operation&gt;</td>
<td>The request operation, such as get a list, search, and download.</td>
</tr>
<tr>
<td>&lt;module&gt;</td>
<td>The API module. For the CM API, the module is: “cm”.</td>
</tr>
<tr>
<td>&lt;object&gt;</td>
<td>The module specific object.</td>
</tr>
<tr>
<td>&lt;object_id&gt;</td>
<td>(Optional) The module specific object ID, if appropriate.</td>
</tr>
</tbody>
</table>

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.

Using Curl

Curl is a multi-platform command-line tool used to transfer data using multiple protocols. This tool is supported on many systems, including Windows, Unix, Linux and Mac. In this document Curl is used in the examples to build CM API requests using the HTTP over SSL (https) protocol, which is required by the Qualys CM API framework.

Want to learn more? Visit http://curl/haxx/se
The following Curl options are used according to different situations:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-u &quot;LOGIN:PASSWORD&quot;</td>
<td>This option is used for basic authentication.</td>
</tr>
<tr>
<td>-X &quot;POST&quot;</td>
<td>This option is used to provide a method other than the default method, GET.</td>
</tr>
<tr>
<td>-H &quot;content-type&quot;</td>
<td>This option is used to provide a custom HTTP request header parameter for content type, to specify the MIME type of the curl’s payload.</td>
</tr>
<tr>
<td>--data-binary</td>
<td>This option is used to specify the POST data. See the examples below.</td>
</tr>
</tbody>
</table>

The sample below shows a typical Curl request using options mentioned above and how they interact with each other. The option -X "POST" tells Curl to execute the request using the HTTP POST method. The option --data-binary @-" tells Curl to read the POST data from its standard input (stdin). The string "< file.xml" is interpreted by the shell to redirect the content of the file to the stdin of the command. The option -H "content-type: text/xml" tells Curl the POST data in “file.xml” is XML in text format.

curl -H "content-type: text/xml" -X "POST" --data-binary @- "https://example.com" < file.xml

This documentation uses Curl examples showing the POST data in the “file.xml” file. This is referred to as Request POST Data. This can also be referred to as the Payload.

**JSON Support**

The Qualys CM API support JSON requests and responses. Learn more
XML Output and Schemas

- Alert XSD
  https://qualysapi.qualys.com/qps/xsd/1.0/cm/alert.xsd
- Profile XSD
  https://qualysapi.qualys.com/qps/xsd/1.0/cm/profile.xsd
- Ruleset XSD
  https://qualysapi.qualys.com/qps/xsd/1.0/cm/ruleset.xsd
- Rule XSD
  https://qualysapi.qualys.com/qps/xsd/1.0/cm/rule.xsd

XML Output Pagination / Truncation Logic

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

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

Request 1:
Search for alerts for the IP address 10.10.30.70. The service request in the POST data file “file.xml” defines this search criteria.

```
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
 --data-binary @-
 "https://qualysapi.qualys.com/qps/rest/1.0/search/cm/alert" <
 file.xml
```

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

Request POST Data for Request 1:

```
<ServiceRequest>
  <preferences>
    <limitResults>5</limitResults>
  </preferences>
  <filters>
    <Criteria field="ipAddress"
```
Chapter 1 — Welcome
Introduction to the CM API Paradigm

operator="EQUALS">10.10.30.70</Criteria>
   </filters>
</ServiceRequest>

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

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

Request 2:
To get the next page of results, you need to edit your service request in “file.xml” that will
be passed to API request as a POST payload. According to the <lastId> element returned
in the first page, you want the next page of results to start with the object ID 124 or
greater.

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST"
--data-binary @-  
"https://qualysapi.qualys.com/qps/rest/1.0/search/cm/alert" <
file.xml

Request POST Data for Request 2:
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).

<ServiceRequest>
   <filters>
      <Criteria field="ipAddress" operator="EQUALS">10.10.30.70</Criteria>
      <Criteria field="id" operator="GREATER">123</Criteria>
   </filters>
</ServiceRequest>
Setting the 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.

<ServiceRequest>
  <filters>
    <Criteria> ... </Criteria>
  </filters>
  <preferences>
    <limitResults>200</limitResults>
  </preferences>
</ServiceRequest>
Base URL to the Qualys API Server

The Qualys API documentation and sample code within it use the API server URL for Qualys US Platform 1: qualysapi.qualys.com.

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

<table>
<thead>
<tr>
<th>Account Location</th>
<th>API Server URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>US Platform 1</td>
<td><a href="https://qualysapi.qualys.com">https://qualysapi.qualys.com</a></td>
</tr>
<tr>
<td>EU Platform</td>
<td><a href="https://qualysapi.qualys.eu">https://qualysapi.qualys.eu</a></td>
</tr>
</tbody>
</table>

Authentication

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 more 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.

Example

Basic authentication - recommended option:

curl -u "USERNAME:PASSWORD"
https://qualysapi.qualys.com/qps/rest/1.0/download/cm/alert

where qualysapi.qualys.com is the base URL to the Qualys API server where your account is located.
Know your Portal Version

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

**URL:** https://qualysapi.qualys.com/qps/rest/portal/version

**Methods allowed:** GET

**Examples**

**Example 1: XML**

**API Request:**

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

**Response:**

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Portal-Version>
      <PortalApplication-VERSION>2.18.0.0-1</PortalApplication-VERSION>
      <WAS-VERSION>4.12.0</WAS-VERSION>
      <CM-VERSION>1.11.0</CM-VERSION>
      <MDS-VERSION>2.10.3</MDS-VERSION>
      <CA-VERSION>1.9.0.0</CA-VERSION>
      <MPS-VERSION>0.4.1</MPS-VERSION>
      <QUESTIONNAIRE-VERSION>2.3.0</QUESTIONNAIRE-VERSION>
      <WAF-VERSION>1.21.0</WAF-VERSION>
    </Portal-Version>
    <QWeb-Version>
      <WEB-VERSION>8.9.2.0-SNAPSHOT-20161214103056#7</WEB-VERSION>
      <SCANNER-VERSION>9.1.14-1</SCANNER-VERSION>
      <VULNSIGS-VERSION>2.3.493-1</VULNSIGS-VERSION>
    </QWeb-Version>
  </data>
</ServiceResponse>
```
Example 2: JSON

API Request:

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

Response:

```json
{
    "ServiceResponse": {
        "data": [
            {
                "Portal-Version": {
                    "PortalApplication-VERSION": "2.18.0.0-1",
                    "WAS-VERSION": "4.12.0",
                    "CM-VERSION": "1.11.0",
                    "MDS-VERSION": "2.10.3",
                    "CA-VERSION": "1.9.0.0",
                    "MPS-VERSION": "0.4.1",
                    "QUESTIONNAIRE-VERSION": "2.3.0",
                    "WAF-VERSION": "1.21.0"
                },
                "QWeb-Version": {
                    "WEB-VERSION": "8.9.2.0-SNAPSHOT-20161214103056#7",
                    "SCANNER-VERSION": "9.1.14-1",
                    "VULNSIGS-VERSION": "2.3.493-1"
                }
            }
        ],
        "responseCode": "SUCCESS",
        "count": 1
    }
}
```
CM API

Use these API functions to download information from your Continuous Monitoring (CM) application.

**Alerts**
- Search alerts
- View details of an alert
- Download alerts

**Profiles**
- Search profiles
- View details of a profile

**Rulesets**
- Search rulesets
- View details of a ruleset

**Rules**
- Search rules
- View details of a rule
Search alerts

Returns a list of alerts in the user’s account.

**URL:** https://qualysapi.qualys.com/qps/rest/1.0/search/cm/alert/

**Methods allowed:** POST

**Input**

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: XML Elements for descriptions.

<table>
<thead>
<tr>
<th>Element</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (Integer)</td>
<td></td>
</tr>
<tr>
<td>isHidden (Boolean)</td>
<td></td>
</tr>
<tr>
<td>eventType (Keyword)</td>
<td></td>
</tr>
<tr>
<td>eventDate (Date)</td>
<td></td>
</tr>
<tr>
<td>ipAddress (Text)</td>
<td></td>
</tr>
<tr>
<td>alertDate (Date)</td>
<td></td>
</tr>
<tr>
<td>hostname (Text)</td>
<td></td>
</tr>
<tr>
<td>profileTitle (Text)</td>
<td></td>
</tr>
</tbody>
</table>

**Allowed Operators**

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>EQUALS, NOT EQUALS, GREATER, LESSER, IN</td>
</tr>
<tr>
<td>Text</td>
<td>CONTAINS, EQUALS, NOT EQUALS</td>
</tr>
<tr>
<td>Date</td>
<td>EQUALS, NOT EQUALS, GREATER, LESSER</td>
</tr>
<tr>
<td>Keyword</td>
<td>EQUALS, NOT EQUALS, IN</td>
</tr>
<tr>
<td>Boolean</td>
<td>(true/false) EQUALS, NOT EQUALS</td>
</tr>
</tbody>
</table>
Example

Return a list of alerts in the user’s account for IP address 10.10.30.70.

Request:
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @- "https://qualysapi.qualys.com/qps/rest/1.0/search/cm/alert/" < file.xml

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

Request POST Data:
<ServiceRequest>
  <filters>
    <Criteria field="ipAddress" operator="EQUALS">10.10.30.70</Criteria>
  </filters>
</ServiceRequest>

Response:
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/xsd/1.0/cm/alert.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <Alert>
      <id>244402</id>
      <source>REMEDIATION</source>
      <eventType>SSL_NEW</eventType>
      <triggerUuid>3d41baf9-7caa-4269-9889-d7377aeeace5</triggerUuid>
      <ipAddress>10.10.30.70</ipAddress>
      <hostname>2k3-sp2-25-69.qualys.com</hostname>
      <eventDate>2014-06-04T10:57:43Z</eventDate>
      <alertDate>2014-06-04T10:57:48Z</alertDate>
      <isHidden>true</isHidden>
      <profile>
        <id>7401</id>
      </profile>
    </Alert>
  </data>
</ServiceResponse>
<title>All Critical</title>
<dateCreated>2013-09-16T19:54:48Z</dateCreated>
<dateUpdated>2013-09-16T19:54:48Z</dateUpdated>
<frequency>FREQ_NEVER</frequency>
<isActive>true</isActive>
<includedIps>10.10.10.1-10.10.31.255</includedIps>
<targetList>10.10.10.1-10.10.31.255</targetList>
</profile>
>alertInfo>
<port>0</port>
<sslName>2k3-sp2-25-69.qualys.com</sslName>
<sslIssuer>2k3-sp2-25-69.qualys.com</sslIssuer>
</alertInfo>
</Alert>
</data>
</ServiceResponse>
View details of an alert

Returns details for an alert. Want to find an alert ID to use as input? See Search alerts.

URL: 
https://qualysapi.qualys.com/qps/rest/1.0/get/cm/alert/<id>
Methods allowed: GET

Input

The element “id” (Integer) is required, where “id” identifies the alert.

Example

View details for the alert with the ID 246213.

Request:
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/1.0/get/cm/alert/246213"

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/1.0/cm/alert.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <Alert>
      <id>246213</id>
      <source>REMEDIATION</source>
      <eventType>HOST_UPDATED</eventType>
      <triggerUuid>3d41baf9-7caa-4269-9889-d7377aeaaace5</triggerUuid>
      <ipAddress>10.10.30.240</ipAddress>
      <hostname>win12-30-240</hostname>
      <eventDate>2014-06-04T18:11:54Z</eventDate>
      <alertDate>2014-06-04T18:11:59Z</alertDate>
      <isHidden>false</isHidden>
      <profile>
        <id>7401</id>
        <title>All Critical</title>
      </profile>
    </Alert>
  </data>
</ServiceResponse>
<alertInfo>
  <operatingSystem>Windows Server 2012 Standard 64 bit Edition</operatingSystem>
  <port>0</port>
</alertInfo>
</Alert>
</data>
</ServiceResponse>
Download alerts

Downloads a list of alerts in CSV or CEF format.

**URL:**  
https://qualysapi.qualys.com/qps/rest/1.0/download/cm/alert/?format=<format>

**Methods allowed:**  
POST

**Input**

The element “format” (Text) is required, where “format” is the file format (csv or cef).

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: XML Elements for descriptions.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (Integer)</td>
<td></td>
<td>EQUALS, NOT EQUALS, GREATER, LESSER, IN</td>
</tr>
<tr>
<td>eventType (Keyword)</td>
<td></td>
<td>EQUALS, NOT EQUALS, IN</td>
</tr>
<tr>
<td>ipAddress (Text)</td>
<td></td>
<td>CONTAINS, EQUALS, NOT EQUALS</td>
</tr>
<tr>
<td>alertDate (Date)</td>
<td></td>
<td>EQUALS, NOT EQUALS, GREATER, LESSER</td>
</tr>
<tr>
<td>hostname (Text)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>profileTitle (Text)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Allowed Operators**

- **Integer:** EQUALS, NOT EQUALS, GREATER, LESSER, IN
- **Text:** CONTAINS, EQUALS, NOT EQUALS
- **Date:** EQUALS, NOT EQUALS, GREATER, LESSER
- **Keyword:** EQUALS, NOT EQUALS, IN
- **Boolean:** (true/false) EQUALS, NOT EQUALS

**Example**

Download alerts for open ports in CEF format.

**Request:**

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @- "https://qualysapi.qualys.com/qps/rest/1.0/download/cm/alert/?format=cef" < file.xml
Chapter 2 — CM API
Download alerts

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

Request POST Data:

```xml
<ServiceRequest>
  <filters>
    <Criteria field="eventType" operator="EQUALS">PORT_OPEN</Criteria>
  </filters>
</ServiceRequest>
```

Sample CEF Output:

```
Jun 06 2014 15:43:9 83306MM.local 10.40.2.210
CEF:0|QUALYS|QualysGuard|CM-1.4|PORT|PORT_OPEN|0|cat=PORT
dhost=2k8core-30-21.2k864sp0.qualys.com dst=10.10.30.21 dmac=NA
dntdom=2K8CORE-30-21 rt=Nov 05 2013 15:57:21
cs1operatingSystem=Windows 2008 Enterprise Server Service Pack 2
dpt=61466 cs2protocol=udp cs3defaultService=NA
```

Sample CSV Output:

```
"AlertId","Event Type","ProfileId","Profile Name","IP Address","Hostname","Operating System","Event Date","Alert Date"
"237124","PORT_OPEN","7401","All Critical","10.10.30.21","2k8core-30-21.2k864sp0.qualys.com","-","2013-12-31 15:57:21","2013-12-31 15:57:22"
"237172","PORT_OPEN","5601","My Profile","10.10.30.21","2k8core-30-21.2k864sp0.qualys.com","-","2013-12-31 15:57:21","2013-12-31 15:57:23"
```
Search profiles

Returns a list of monitoring profiles in the user’s account.

**URL:**  
https://qualysapi.qualys.com/qps/rest/1.0/search/cm/profile/

**Methods allowed:**  
POST

**Input**

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: XML Elements for descriptions.

<table>
<thead>
<tr>
<th>Allowed Operators</th>
<th>Integer</th>
<th>EQUALS, NOT_EQUALS, GREATER, LESSER, IN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>CONTAINS, EQUALS, NOT_EQUALS</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>EQUALS, NOT_EQUALS, GREATER, LESSER</td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>EQUALS, NOT_EQUALS, IN</td>
<td></td>
</tr>
<tr>
<td>Boolean</td>
<td>(true/false) EQUALS, NOT_EQUALS</td>
<td></td>
</tr>
</tbody>
</table>

**Examples**

Return a list of profiles that use a ruleset with the word "critical" in the ruleset title.

**Request:**

curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @- "https://qualysapi.qualys.com/qps/rest/1.0/search/cm/profile/" < file.xml

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

```xml
<ServiceRequest>
  <filters>
    <Criteria field="ruleSetTitle" operator="CONTAINS">critical</Criteria>
  </filters>
</ServiceRequest>
```

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/1.0/cm/profile.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <Profile>
      <id>7401</id>
      <title>All Critical</title>
      <uuid>d7af450c-828c-4101-a653-737f10d596c6</uuid>
      <dateCreated>2013-09-16T19:54:48Z</dateCreated>
      <dateUpdated>2013-09-16T19:54:48Z</dateUpdated>
      <frequency>FREQ_NEVER</frequency>
      <isActive>true</isActive>
      <includedIps>10.10.10.1-10.10.31.255</includedIps>
      <targetList>10.10.10.1-10.10.31.255</targetList>
      <ruleset>
        <id>4001</id>
        <title>All Critical</title>
        <description>Critical security risks to be addressed immediately.</description>
        <dateCreated>2013-09-16T19:36:10Z</dateCreated>
        <dateUpdated>2013-09-16T19:36:10Z</dateUpdated>
        <isTemplate>false</isTemplate>
      </ruleset>
    </Profile>
  </data>
</ServiceResponse>
```
View details of a profile

Returns details for a monitoring profile. Want to find a profile ID to use as input? See Search profiles.

**URL:**
https://qualysapi.qualys.com/qps/rest/1.0/get/cm/profile/<id>

**Methods allowed:** GET

**Input**
The element “id” (Integer) is required, where “id” identifies the profile.

**Example**
View details for the profile with the ID 7401.

**Request:**
curl -u "USERNAME:PASSWORD" "https://qualysapi.qualys.com/qps/rest/1.0/get/cm/profile/7401"

**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/1.0/cm/profile.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Profile>
            <id>7401</id>
            <title>All Critical</title>
            <uuid>d7af450c-828c-4101-a653-737f10d596c6</uuid>
            <dateCreated>2013-09-16T19:54:48Z</dateCreated>
            <dateUpdated>2013-09-16T19:54:48Z</dateUpdated>
            <frequency>FREQ_NEVER</frequency>
            <isActive>true</isActive>
            <includedIps>10.10.10.1-10.10.31.255</includedIps>
            <targetList>10.10.10.1-10.10.31.255</targetList>
        </Profile>
    </data>
</ServiceResponse>
```
Chapter 2 — CM API
View details of a profile

<description>Critical security risks to be addressed immediately.</description>
<dateCreated>2013-09-16T19:36:10Z</dateCreated>
<dateUpdated>2013-09-16T19:36:10Z</dateUpdated>
<isTemplate>false</isTemplate>
</ruleset>
</Profile>
</data>
</ServiceResponse>
Search rulesets

Returns a list of rulesets in the user’s account.

**URL:**
https://qualysapi.qualys.com/qps/rest/1.0/search/cm/ruleset/

**Methods allowed:** POST

**Input**

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: XML Elements for descriptions.

<table>
<thead>
<tr>
<th>Element</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Integer</td>
</tr>
<tr>
<td>dateCreated</td>
<td>Date</td>
</tr>
<tr>
<td>title</td>
<td>Text</td>
</tr>
<tr>
<td>dateUpdated</td>
<td>Date</td>
</tr>
<tr>
<td>description</td>
<td>Text</td>
</tr>
</tbody>
</table>

**Allowed Operators**

- **Integer:** EQUALS, NOT EQUALS, GREATER, LESSER, IN
- **Text:** CONTAINS, EQUALS, NOT EQUALS
- **Date:** EQUALS, NOT EQUALS, GREATER, LESSER
- **Keyword:** EQUALS, NOT EQUALS, IN
- **Boolean:** (true/false) EQUALS, NOT EQUALS

**Examples**

Return a list of rulesets with the word "critical" in the title.

**Request:**
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @- "https://qualysapi.qualys.com/qps/rest/1.0/search/cm/ruleset/" < file.xml

Note: “file.xml” contains the request POST data.
Chapter 2 — CM API
Search rulesets

Request POST Data:

```
<ServiceRequest>
  <filters>
    <Criteria field="title" operator="CONTAINS">critical</Criteria>
  </filters>
</ServiceRequest>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="https://qualysapi.qualys.com/qps/
  xsd/1.0/cm/ruleset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <RuleSet>
      <id>4001</id>
      <title>All Critical</title>
      <description>Critical security risks to be addressed immediately.</description>
      <dateCreated>2013-09-16T19:36:10Z</dateCreated>
      <dateUpdated>2013-09-16T19:36:10Z</dateUpdated>
      <isTemplate>false</isTemplate>
      <rule>
        <list>
          <Rule>
            <id>6001</id>
            <ruleType>HOST</ruleType>
            <eventTypes>HOST_FOUND, HOST_UPDATED, HOST_PURGED</eventTypes>
            <jsonData>{"ruleType":"HOST","eventTypes":["HOST_FOUND","HOST_UPDATED","HOST_PURGED"],"criteria":[]},"uiState":{"eventType.HOST_FOUND":"on","eventType.HOST_UPDATED":"on","eventType.HOST_PURGED":"on","operatingSystemType":","operatingSystemValue":","hostnameType":","hostnameValue":","netbiosNameType":","netbiosNameValue":}}
          </Rule>
        </list>
      </rule>
    </RuleSet>
  </data>
</ServiceResponse>
```
Chapter 2 — CM API
Search rulesets

onData>
  <dateCreated>2013-09-17T15:32:10Z</dateCreated>
</Rule>

<Rule>
  <id>6002</id>
  <ruleType>VULN</ruleType>
  <eventTypes>VULN_OPEN, VULN_CLOSED, VULN_ACTIVE, VULN_REOPENED</eventTypes>

  <jsonData>{"ruleType": "VULN", "eventTypes": ["VULN_OPEN", "VULN_CLOSED", "VULN_ACTIVE", "VULN_REOPENED"], "criteria": [], "uiState": {"eventType.VULN_OPEN": "on", "eventType.VULN_CLOSED": "on", "eventType.VULN_ACTIVE": "on", "eventType.VULN_REOPENED": "on", "titleType": "", "titleValue": "", "qidType": "", "qidValue": "", "cveType": "", "cveValue": ""}}</jsonData>

  <dateCreated>2013-09-17T15:32:11Z</dateCreated>
</Rule>
</list>
</RuleSet>
</data>
</ServiceResponse>
View details of a ruleset

Returns details for a ruleset. Want to find a ruleset ID to use as input? See Search rulesets.

**URL:**
https://qualysapi.qualys.com/qps/rest/1.0/get/cm/ruleset/<id>

**Methods allowed:** GET

**Input**

The element “id” (Integer) is required, where “id” identifies the ruleset.

**Example**

View details for the ruleset with the ID 4001.

**Request:**
curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/1.0/get/cm/ruleset/4001"

**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/1.0/cm/ruleset.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>1</count>
  <data>
    <RuleSet>
      <id>4001</id>
      <title>All Critical</title>
      <description>Critical security risks to be addressed immediately.</description>
      <dateCreated>2013-09-16T19:36:10Z</dateCreated>
      <dateUpdated>2013-09-16T19:36:10Z</dateUpdated>
      <isTemplate>false</isTemplate>
      <rule>
        <list>
          <Rule>
            <id>6001</id>
            <ruleType>HOST</ruleType>
            <eventTypes>HOST_FOUND, HOST_UPDATED,
HOST_PURGED</eventTypes>

<jsonData>{"ruleType": "HOST", "eventTypes": ["HOST_FOUND", "HOST_UPDATED", "HOST_PURGED"], "criteria": [], "uiState": {"eventType.HOST_FOUND": "on", "eventType.HOST_UPDATED": "on", "eventType.HOST_PURGED": "on", "operatingSystemType": "", "operatingSystemValue": "", "hostnameType": "", "hostnameValue": "", "netbiosNameType": "", "netbiosNameValue": ""}}</jsonData>

<dateCreated>2013-09-17T15:32:10Z</dateCreated></Rule>

<Rule>
 <id>6002</id>
 <ruleType>VULN</ruleType>
 <eventTypes>VULN_OPEN, VULN_CLOSED, VULN_ACTIVE, VULN_REOPENED</eventTypes>

<jsonData>{"ruleType": "VULN", "eventTypes": ["VULN_OPEN", "VULN_CLOSED", "VULN_ACTIVE", "VULN_REOPENED"], "criteria": [], "uiState": {"eventType.VULN_OPEN": "on", "eventType.VULN_CLOSED": "on", "eventType.VULN_ACTIVE": "on", "eventType.VULN_REOPENED": "on", "titleType": "", "titleValue": "", "qidType": "", "qidValue": "", "cveType": "", "cveValue": ""}}</jsonData>

<dateCreated>2013-09-17T15:32:11Z</dateCreated></Rule>
</list>
</RuleSet>
</data>
</ServiceResponse>
Chapter 2 — CM API

Search rules

Search rules returns a list of rules (which are part of rule sets) in the user’s account.

**URL:**
https://qualysapi.qualys.com/qps/rest/1.0/search/cm/rule/

**Methods allowed:**
POST

**Input**

Allowed input elements are listed below. The associated data type for each element appears in parentheses. These elements are optional and act as filters. When multiple elements are specified, parameters are combined using a logical AND. All dates must be entered in UTC date/time format. See Reference: XML Elements for descriptions.

- **id (Integer)**
- **ruleType (Keyword)**

**Allowed Operators**

<table>
<thead>
<tr>
<th>Type</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>EQUALS, NOT EQUALS, GREATER, LESSER, IN</td>
</tr>
<tr>
<td>Text</td>
<td>CONTAINS, EQUALS, NOT EQUALS</td>
</tr>
<tr>
<td>Date</td>
<td>EQUALS, NOT EQUALS, GREATER, LESSER</td>
</tr>
<tr>
<td>Keyword</td>
<td>EQUALS, NOT EQUALS, IN</td>
</tr>
<tr>
<td>Boolean</td>
<td>(true/false) EQUALS, NOT EQUALS</td>
</tr>
</tbody>
</table>

**Examples**

Returns a list of rules related to hosts - new hosts, updated hosts, purged hosts, etc.

**Request:**

```bash
curl -u "USERNAME:PASSWORD" -H "content-type: text/xml" -X "POST" --data-binary @- "https://qualysapi.qualys.com/qps/rest/1.0/search/cm/rule/" < file.xml
```

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

---

Qualys Continuous Monitoring API
Request POST Data:

```xml
<ServiceRequest>
  <filters>
    <Criteria field="ruleType" operator="EQUALS">HOST</Criteria>
  </filters>
</ServiceRequest>
```

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/1.0/cm/rule.xsd">
  <responseCode>SUCCESS</responseCode>
  <count>19</count>
  <hasMoreRecords>false</hasMoreRecords>
  <data>
    <Rule>
      <id>1413</id>
      <ruleType>HOST</ruleType>
      <eventTypes>HOST_FOUND, HOST_UPDATED</eventTypes>
      <jsonData>"{"ruleType": "HOST", "eventTypes": ["HOST_FOUND", "HOST_UPDATED"], "criteria": [{"propertyName": "name", "propertyValue": "blah", "expressionType": "EQUALS"}, {"propertyName": "operatingSystem", "propertyValue": "blah", "expressionType": "EQUALS"}, {"propertyName": "hostname", "propertyValue": "blah", "expressionType": "EQUALS"}]}
      <dateCreated>2013-09-05T16:21:00Z</dateCreated>
      <criteria>
        <list>
          <RuleCriteria>
            <id>1437</id>
            <propertyName>name</propertyName>
            <propertyValue>blah</propertyValue>
            <expressionType>EQUALS</expressionType>
          </RuleCriteria>
          <RuleCriteria>
            <id>1439</id>
          </RuleCriteria>
        </list>
      </criteria>
    </Rule>
  </data>
</ServiceResponse>
```
<propertyName>operatingSystem</propertyName>
<propertyValue>blah</propertyValue>
<expressionType>EQUALS</expressionType>
</RuleCriteria>
</list>
</criteria>
</Rule>
</data>
</ServiceResponse>
View details of a rule

Returns details for a rule. Want to find a rule ID to use as input? See Search rules.

**URL:**
https://qualysapi.qualys.com/qps/rest/1.0/get/cm/rule/<id>

**Methods allowed:**
GET

**Input**

The element “id” (Integer) is required, where “id” identifies the rule.

**Example**

View details for the rule with the ID 6002.

**Request:**

curl -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/qps/rest/1.0/get/cm/rule/6002"

**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/1.0/cm/rule.xsd">
    <responseCode>SUCCESS</responseCode>
    <count>1</count>
    <data>
        <Rule>
            <id>6002</id>
            <ruleType>VULN</ruleType>
            <eventTypes>VULN_OPEN, VULN_CLOSED, VULN_ACTIVE, VULN_REOPENED</eventTypes>
            <jsonData>{"ruleType":"VULN","eventTypes": ["VULN_OPEN","VULN_CLOSED","VULN_ACTIVE","VULN_REOPENED"],"criteria":[]},
            "uiState": {"eventType.VULN_OPEN": "on", "eventType.VULN_CLOSED": "on", "eventType.VULN_ACTIVE": "on", "eventType.VULN_REOPENED": "on","titleType": "","titleValue": "","qidType": "","qidValue": ""}</jsonData>
        </Rule>
    </data>
</ServiceResponse>
```
Chapter 2 — CM API

View details of a rule

```xml
<jsonData>
  <dateCreated>2013-09-17T15:32:11Z</dateCreated>
  <criteria>
    <list/>
  </criteria>
</Rule>
</data>
</ServiceResponse>
```
### Reference: XML Elements

<table>
<thead>
<tr>
<th>Element (data type)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alert</strong></td>
<td></td>
</tr>
<tr>
<td>id (Integer)</td>
<td>The alert ID. This element is assigned by the service and is required for a get alert request.</td>
</tr>
<tr>
<td>eventType (Keyword)</td>
<td>The type of event that triggered the alert: HOST_FOUND, HOST_UPDATED, HOST_PURGED, PORT_OPEN, PORT_CHANGED, PORT_CLOSED, SOFTWARE_ADDED, SOFTWARE_REMOVED, SSL_NEW, SSL_EXPIRED, SSL_EXPIRY, TICKET_OPEN, TICKET_RESOLVED, TICKET_CLOSED, VULN_OPEN, VULN_CLOSED, VULN_REOPENED, VULN_ACTIVE, VULN_PREDICTION_ADDED, VULN_PREDICTION_CHANGED, VULN_PREDICTION_CLOSED</td>
</tr>
<tr>
<td>ipAddress (Text)</td>
<td>The impacted host’s IP address.</td>
</tr>
<tr>
<td>hostname (Text)</td>
<td>The impacted host’s hostname.</td>
</tr>
<tr>
<td>isHidden (Boolean)</td>
<td>Is the alert hidden from view? (true or false).</td>
</tr>
<tr>
<td>eventDate (Date)</td>
<td>The date of the event that triggered the alert in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
<tr>
<td>alertDate (Date)</td>
<td>The date of the alert in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
<tr>
<td>profileTitle (Text)</td>
<td>The name of the monitoring profile that resulted in the alert.</td>
</tr>
<tr>
<td><strong>Profile</strong></td>
<td></td>
</tr>
<tr>
<td>id (Integer)</td>
<td>The monitoring profile ID. This element is assigned by the service and is required for a get profile request.</td>
</tr>
<tr>
<td>title (Text)</td>
<td>The name of the monitoring profile.</td>
</tr>
<tr>
<td>uuid (Integer)</td>
<td>The monitoring profile UUID. This element is assigned by the service.</td>
</tr>
<tr>
<td>frequency (Keyword)</td>
<td>The notification frequency setting defined in the profile: FREQ_NEVER, FREQ_5_MINUTES, FREQ_20_MINUTES, FREQ_1_HR, FREQ_2_HRS, FREQ_6_HRS, FREQ_12_HRS, FREQ_WEEKLY, FREQ_DAILY</td>
</tr>
<tr>
<td>isActive (Boolean)</td>
<td>Is the monitoring profile active? (true or false).</td>
</tr>
<tr>
<td>ruleSetTitle (Text)</td>
<td>The name of the ruleset assigned to the monitoring profile.</td>
</tr>
</tbody>
</table>
### Ruleset

<table>
<thead>
<tr>
<th>Element (data type)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (Integer)</td>
<td>The ruleset ID. This element is assigned by the service and is required for a get ruleset request.</td>
</tr>
<tr>
<td>title (Text)</td>
<td>The name of the ruleset.</td>
</tr>
<tr>
<td>description (Text)</td>
<td>The user-provided description for the ruleset.</td>
</tr>
<tr>
<td>dateCreated (Date)</td>
<td>The date the ruleset was created in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
<tr>
<td>dateUpdated (Date)</td>
<td>The date the ruleset was last updated in UTC date/time format (YYYY-MM-DDTHH:MM:SSZ).</td>
</tr>
</tbody>
</table>

### Rule

<table>
<thead>
<tr>
<th>Element (data type)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id (Integer)</td>
<td>The rule ID. This element is assigned by the service and is required for a get rule request.</td>
</tr>
<tr>
<td>ruleType (Keyword)</td>
<td>The type of rule: HOST, VULN, PORT, SSL, SW</td>
</tr>
</tbody>
</table>
# Error Messages

This appendix describes the types of error messages returned from CM API requests.

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<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>headers: Length of all headers cannot exceed 2048 characters.</td>
<td>The values of all headers cannot exceed 2048 characters.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria: Field is required.</td>
</tr>
<tr>
<td>Criteria: Invalid criteria (&lt;field name&gt;).</td>
</tr>
<tr>
<td>Criteria: Invalid operator for criteria '&lt;field&gt;' (&lt;operator&gt;).</td>
</tr>
<tr>
<td>Criteria: Value is required for criteria '&lt;field&gt;'.</td>
</tr>
<tr>
<td>Criteria: Invalid value format for criteria '&lt;field&gt;': &lt;value&gt;.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>You are not authorized to access the application through the API.</td>
</tr>
</tbody>
</table>
# Appendix A — Error Messages

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>You do not have access to the module Continuous Monitoring required by this API.</td>
<td>Please contact your account manager to have CM enabled in your subscription.</td>
</tr>
<tr>
<td>No data shall be passed for this operation.</td>
<td>The POST request does not specify a data element.</td>
</tr>
<tr>
<td>User is not authorized to perform this operation on specified object(s).</td>
<td>You must be granted access to these objects in your user scope.</td>
</tr>
<tr>
<td>Operation %s does not support search filters.</td>
<td>Do not provide search filters for this operation.</td>
</tr>
</tbody>
</table>
JSON Support

The Qualys CM API supports JSON requests and responses. Samples are shown below.

**Headers used in samples**

<table>
<thead>
<tr>
<th>Send JSON request</th>
<th>&quot;Content-Type: application/json&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get response in JSON</td>
<td>&quot;Accept: application/json&quot;</td>
</tr>
</tbody>
</table>

**Example 1: Get profile**

**Request:**

```
curl -s -k -H "Accept: application/json" -n -u "acme_ss2:passwd" https://qualysapi.qualys.com/qps/rest/1.0/get/cm/profile/95001
```

**Response:**

```
{
   "ServiceResponse": { 
       "data": [ 
           { 
               "Profile": { 
                   "title": "NEW-Profile-12144557", 
                   "dateCreated": "2015-11-12T22:46:10Z", 
                   "ruleset": { 
                       "dateUpdated": "2015-11-12T22:45:35Z", 
                       "isTemplate": "false", 
                       "title": "NEW-CM-RULE-003-12144533", 
                       "description": "Smoke test", 
                       "id": 188202, 
                       "dateCreated": "2015-11-12T22:45:35Z" 
                   }, 
                   "targetList": "10.10.26.252" 
               } 
           } 
       ] 
   } 
}
```
Example 2: Search for rule

Request:

```bash
```

POST data:

```json
{
   "ServiceRequest": {
      "filters": {
         "Criteria": [{
            "field": "id",
            "operator": "EQUALS",
            "value": "179201"
         },{
            "field": "ruleType",
            "operator": "EQUALS",
            "value": "HOST"
         }]
      }
   }
}
```

Response:

```json
{
   "ServiceResponse": {
      "data": [
         {
            "Rule": {
               "id": 179201,
               "ruleType": "HOST",
               "dateCreated": "2015-09-30T20:15:29Z",
```
"jsonData": {
    "ruleType": "HOST",
    "eventTypes": ["HOST_FOUND"],
    "criteria": [],
    "uiState": {
        "eventType.HOST_FOUND": "on",
        "operatingSystemType": "",
        "operatingSystemValue": "",
        "hostnameType": "",
        "hostnameValue": "",
        "netbiosNameType": "",
        "netbiosNameValue": ""
    }
},

"eventTypes": "HOST_FOUND",
"criteria": {
    "list": []
},

"count": 1,
"hasMoreRecords": "false",
"responseCode": "SUCCESS"}