Container Runtime Security

API Guide
Version 1.6

September 25, 2020

Qualys, Inc.
919 E Hillsdale Blvd
4th Floor
Foster City, CA 94404
1 (650) 801 6100

Table of Contents

About this Guide

This user guide is intended for application developers who will use the APIs for Container Runtime Security (CRS). CRS a separately licensed feature within the Qualys Container Security (CS) product.

CRS provides runtime visibility and protection for containers. This is achieved by instrumenting images with Container Security components that gather functional-level behavioral data about the processes running within a container. This behavioral data is used by Container Security to visualize process activity. You can create and apply security policies that provide custom security controls based on the container's activity.

CRS is not enabled by default for existing or new customers. If you're interested in this feature, please contact your Qualys Account Manager or Qualys Support.

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 apps help businesses 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, Deutsche Telekom, Fujitsu, HCL, HP Enterprise, IBM, Infosys, NTT, Optiv, SecureWorks, Tata Communications, Verizon and Wipro. The company is also founding member of the Cloud Security Alliance (CSA). For more information, please visit www.qualys.com

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 online support information at www.qualys.com/support/.

Accessing the APIs

Several features of Container Runtime Security are available through REST APIs.

Permissions required to use APIs

- User must have the Container Security (CS) module enabled

- User must have Container Runtime Security (CRS) feature enabled

- User must have API ACCESS permission

Qualys API URLs

Container Security supports the Qualys API gateway for API requests.

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

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

Authentication for Gateway URLs

You must authenticate to the Qualys Cloud Platform using Qualys account credentials (user name and password) and get the JSON Web Token (JWT) before you can start using the Gateway URLs. Use the Qualys Authentication API to get the JWT.

For example:

```
curl -X POST 'https://gateway.qg1.apps.qualys.com/auth' -H 'Content-Type:
application/x-www-form-urlencoded' --data-urlencode 'username=Value' --
data-urlencode 'password=Value' --data-urlencode 'token=true' --data-
urlencode 'permissions=true'
```

where gateway.qg1.apps.qualys.com is the base URL to the Qualys API server where your account is located.

- **username** and **password** are the credentials of the user account for Container Security

- **token** should be true

- **Content-Type** should be "application/x-www-form-urlencoded"

The Authentication API returns a JSON Web Token (JWT) which you can use for authentication during Container Security API calls. The token expires in 4 hours. You must regenerate the token to continue using the Container Security API.

Online API Guide

You can directly access an online API guide from the following URL

```
http://<QualysGatewayURL>/apidocs/csapi/v1.2/runtime
```

For example, if your account is on US Platform 1

```
https://gateway.qg1.apps.qualys.com/apidocs/csapi/v1.2/runtime
```

Configurations

Here is the list of the APIs we currently support for instrumentation configurations:

API Objective	Operator	API Path
Get all configurations in your account	GET	/csapi/v1.2/runtime/configs
Get details for a specific configuration	GET	/csapi/v1.2/runtime/configs/{configId}
Create a new configuration	POST	/csapi/v1.2/runtime/configs
Update a configuration	PUT	/csapi/v1.2/runtime/configs/{configId}

Samples for various operations on configurations:

Get all configurations in your account

/csapi/v1.2/runtime/configs

[GET]

API request:

```
curl 'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/configs' --
header 'Authorization: Bearer <token>'
```

Response:


```
[
    {
        "ID": "5e18c86e4e08ce0001368941",
        "DateCreated": "2020-01-10T18:54:38.82Z",
        "DateUpdated": "2020-01-10T18:54:38.82Z",
        "MQ": "https://cmsqagpublic.qg1.apps.qualys.com/crs/v1.2",
        "PolicyID": "5e18c86e4e08ce0001368940",
        "LogMode": 3,
        "Sniffing": false,
        "Default": true,
        "Name": "default config"
    },
    {
        "ID": "5e1ae506b06e090001bf8741",
        "DateCreated": "2020-01-10T18:54:38.82Z",
        "DateUpdated": "2020-06-04T13:03:53.676Z",
        "MQ": "https://cmsqagpublic.qg1.apps.qualys.com/crs/v1.2",
```

```
            "PolicyID": "5e2587fd6bee780001c5625e",
            "LogMode": 3,
            "Sniffing": false,
            "Default": false,
            "Name": ""
        },
    ...
    ]
```

Get details for a specific configuration

/csapi/v1.2/runtime/configs/{configId}

[GET]

Input Parameters:

Parameter	Description
configId	(Required) Specify the ID of the configuration you want to get details on.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/configs/5e284a064
b80630001437f4e' \
--header 'Authorization: Bearer <token>'
```

Response:

```
{
    "ID": "5e284a064b80630001437f4e",
    "DateCreated": "2020-06-08T10:03:43.507Z",
    "DateUpdated": "2020-06-08T10:07:44.249Z",
    "MQ": "https://cmsqagpublic.qg1.apps.qualys.com/crs/v1.2",
    "PolicyID": "5e18c86e4e08ce0001368940",
    "LogMode": 15,
    "Sniffing": true,
    "Default": false,
    "Name": "example configuration"
}
```

Create a configuration

/csapi/v1.2/runtime/configs

[POST]

Input Parameters:

Parameter	Description
Name	Specify a name for the new configuration.
PolicyID	Specify the ID of the security policy for this container. Sample value: 59c2dc5dc07f870001548489
Sniffing	Set to false by default. Specify true to enable basic network monitoring for malicious activity and network behavior recording.
Default={value}	Set to false by default. Specify true to make this the default configuration for group.

API request:

```
curl --location --request POST
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/configs' \
--header 'Authorization: Bearer <token>'
--header 'Content-Type: text/plain' \
--data-raw '{
  "Name": "example configuration",
  "PolicyID": "59c2dc5dc07f870001548489",
  "Sniffing": true,
  "Default": false
}'
```

Response:

```
{
    "ID": "5ede0cfff42b100001905d58",
    "DateCreated": "2020-06-08T10:03:43.507Z",
    "DateUpdated": "2020-06-08T10:03:43.507Z",
    "MQ": "amqps://username:password@rabbitmq:5671",
    "PolicyID": "59c2dc5dc07f870001548489",
    "LogMode": 15,
    "Sniffing": true,
    "Default": false,
    "Name": "example configuration"
}
```

Update a configuration

/csapi/v1.2/runtime/configs/{configId}

[PUT]

Input Parameters:

Parameter	Description
configId	(Required) The ID of the configuration to update.
Name	Name of the configuration
PolicyID	Specify the ID of the security policy for this container. Sample value: 59c2dc5dc07f870001548489

API request:

```
curl --location --request PUT
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/configs/5ede0cfff
42b100001905d58' \
--header 'Authorization: Bearer <token>'\
--header 'Content-Type: text/plain' \
--data-raw '{
  "Name": "example configuration",
  "PolicyID": "59c2dc5dc07f870001548489",
}'
```

Response:

```
{
    "ID": "5ede0cfff42b100001905d58",
    "DateCreated": "2020-06-08T10:03:43.507Z",
    "DateUpdated": "2020-06-08T10:07:44.249Z",
    "MQ": "amqps://username:password@rabbitmq:5671",
    "PolicyID": "5e18c86e4e08ce0001368940",
    "LogMode": 15,
    "Sniffing": true,
    "Default": false,
    "Name": "example configuration"
}
```

Containers

Here is the list of the APIs we currently support for containers:

API Objective	Operator	API Path
Get runtime details of container	GET	/csapi/v1.2/runtime/containers/{containerSha}
Get runtime profile for container	GET	/csapi/v1.2/runtime/containers/{containerSha}/runtimeprofile
Build a security policy based on a container's behavior	POST	/csapi/v1.2/runtime/containers/{containerSha}/template
Assign instrumentation configuration to container	POST	/csapi/v1.2/runtime/containers/{containerSha}/configs/{configId}

Samples for various operations on containers:

Get runtime details of a container
Get runtime profile for a container
Build a security policy based on a container's behavior
Assign a configuration to a container

Get runtime details of a container

/csapi/v1.2/runtime/containers/{containerSha}

[GET]

Input Parameters:

Parameter	Description
containerSha={value}	(Required) Specify the SHA value of the container for which you want to get runtime details.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/containers/7113e5
aa32875169d41d168a871ca17a510663a6c0ea0e3a9ba03d0eea00cff6' \
--header 'Authorization: Bearer <token>'
```

Response:

```
{
    "ContainerSHA":
"7113e5aa32875169d41d168a871ca17a510663a6c0ea0e3a9ba03d0eea00cff6",
    "ConfigID": "5e7df4f14b89300001cde5cb",
    "Status": "running",
}
```

Get runtime profile for a container

/csapi/v1.2/runtime/containers/{containerSha}/runtimeprofile

[GET]

Input Parameters:

Parameter	Description
containerSha={value}	(Required) Specify the SHA value of the container for which you want to get the runtime profile.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/containers/7113e5
aa32875169d41d168a871ca17a510663a6c0ea0e3a9ba03d0eea00cff6/runtimeprofile
' \
--header 'Authorization: Bearer <token>'
```

Response:


```
{
    "Files": [
        "/etc/ld.so.cache",
        "/etc/resolv.conf",
        "/lib/x86_64-linux-gnu/libacl.so.1",
        "/lib/x86_64-linux-gnu/libacl.so.1.1.0",
        "/lib/x86_64-linux-gnu/libattr.so.1",
        "/lib/x86_64-linux-gnu/libattr.so.1.1.0",
        "/lib/x86_64-linux-gnu/libc-2.19.so",
        "/lib/x86_64-linux-gnu/libc.so.6",
        "/lib/x86_64-linux-gnu/libdl-2.19.so",
        "/lib/x86_64-linux-gnu/libdl.so.2",
        "/lib/x86_64-linux-gnu/libpcre.so.3",
        "/lib/x86_64-linux-gnu/libpcre.so.3.13.1",
        "/lib/x86_64-linux-gnu/libpthread-2.19.so",
        "/lib/x86_64-linux-gnu/libpthread.so.0",
        "/lib/x86_64-linux-gnu/libselinux.so.1"
    ],
    "Programs": [
        "/bin/cat",
        "/bin/ls",
        "/bin/sh"
    ],
    "Ports": null,
    "IPs": null
}
```

Build a security policy based on a container's behavior

/csapi/v1.2/runtime/containers/{containerSha}/template

[POST]

Input Parameters:

Parameter	Description
containerSha	(Required) Specify the SHA value of the container for which you want to create a new custom security policy based on the recorded activities of the specified container.

API request:

```
curl --location --request POST
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/containers/7113e5
aa32875169d41d168a871ca17a510663a6c0ea0e3a9ba03d0eea00cff6/template' \
--header 'Authorization: Bearer <token>'
```

Response:

```
{
    "TemplateID": "5ede15a34b23720001a75560"
}
```

Assign a configuration to a container

/csapi/v1.2/runtime/containers/{containerSha}/configs/{configId}

[POST]

Input Parameters:

Parameter	Description
containerSha	(Required) Specify the SHA value of the container that you're assigning the configuration to.
configId	(Required) Specify the ID of the configuration you want to assign to the container.

API request:

```
curl --location --request POST
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/containers/7113e5
aa32875169d41d168a871ca17a510663a6c0ea0e3a9ba03d0eea00cff6/configs/5e7df4
f14b89300001cde5cb' \
--header 'Authorization: Bearer <token>'
```

Response:

```
response code 200
```

Images

Here is the list of the APIs we currently support for images:

API Objective	Operator	API Path
Instrument image with Qualys instrumentation	POST	/csapi/v1.1/images/{imageId}/instrument
Get CRS configuration of an image with instrumentation	GET	/csapi/v1.2/runtime/images/{imageSha}/agentconfig
Assign instrumentation configuration to an image	POST	/csapi/v1.2/runtime/images/{imageSha}/configs/{configId}

Samples for various operations on images:

Instrument image with Qualys instrumentation
Get CRS configuration of an image with instrumentation
Assign configuration to an image

Instrument image with Qualys instrumentation

Once the instrumenter service is up and running in your environment, you can instrument your images. Note that you can only instrument images that have been scanned by a registry scan job (registry sensor). For this API endpoint, you'll use the Container Security API. To learn more about using Container Security APIs, please refer to the Container Security API User Guide.

/csapi/v1.1/images/{imageId}/instrument

[POST]

Input Parameters:

Parameter	Description
imageId	(Required) Specify the ID or SHA value of the image that you want to instrument.
pullRegistryUuid	The UUID of the registry where the image is located.
pullRepository	Name of the repository where the image is located.
pullTag	Tag associated with the image.
pushRegistryUuid	The UUID of the registry where you want to put the instrumented image.
pushRepository	Name of the repository where you want to put the instrumented image.
pushTag	Tag to be associated with the instrumented image.

API request:

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept:
text/plain' --header 'Authorization: Basic c2x5d3Jfcm06UWF0ZW1wJDEyMw=='
-d '{ \ "imageId": "b5e5ffb5cdea", \ "pullRegistryUuid": "44a38be9-5eaa-
4062-9ee3-5c60c50b430f", \ "pullRepository": "centos", \ "pullTag": "6", \
"pushRegistryUuid": "44a38be9-5eaa-4062-9ee3-5c60c50b430f", \
"pushRepository": "centos", \ "pushTag": "6-1.2" \ }'
'http://qualysapi.qualys.com/csapi/v1.1/images/b5e5ffb5cdea/instrument'
```

Response:

```
response code 200
```

Get CRS configuration of an image with instrumentation

/csapi/v1.2/runtime/images/{imageSha}/agentconfig

[GET]

Input Parameters:

Parameter	Description
imageSha	(Required) Specify the SHA value of the image for which you want to get the CRS agent configuration.
raw	Default value is false. Specify true to get the response format output in raw format.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/images/d6c9104258
01a703530b92f943575b8ea9daa520f77f96e891993f1549a27073/agentconfig' \
--header 'Authorization: Bearer <token>'
```

Response:

```
"0,8,\n---
\n1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,\n---
```

```
\n0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\n---
\nModification_resolvconf_deny,*,3,1,/etc/resolv.conf,,,\n"
```


Assign configuration to an image

/csapi/v1.2/runtime/images/{imageSha}/configs/{configId}

[POST]

Input Parameters:

Parameter	Description
imageSha	(Required) Specify the SHA value of the image for which you want to assign the CRS agent configuration.
configId	(Required) Specify the ID of the configuration you want to assign to the specified image.

API request:

```
curl --location --request POST
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/images/d6c9104258
01a703530b92f943575b8ea9daa520f77f96e891993f1549a27073/configs/5e7df4f14b
89300001cde5cb' \
--header 'Authorization: Bearer <token>'
```

Response:

```
response code 200
```

Events

Here is the list of the APIs we currently support for events:

API Objective	Operator	API Path
Get all events	GET	/csapi/v1.2/runtime/events

Samples for various operations on events:

Get all events in your account

Get all events in your account

/csapi/v1.2/runtime/events

[GET]

There are several options for filtering the events returned in the output. For example, you can only get events created after a certain date, before a certain date or within a date range. You can also filter the list to get events for a particular container or with a certain action type. See all options below.

Input Parameters:

Parameter	Description
eventType	(Required) Specify the type of logs you want to return. Possible values are: STANDARD, BEHAVIOR.
startTime	Specify a starting date/time to get events created after this date. Specify the date in the format ['YYYY'-'MM'-'DD'T'hh':'mm':'ss'].
endTime	Specify an ending date/time to get events created before this date. Specify the date in the format ['YYYY'-'MM'-'DD'T'hh':'mm':'ss'].
filter	Specify a string value for a search query to filter the list of events returned in the output. In the search query you can include any value that appears in the response body like action, system, systemCall, containerSha, uuid, etc. For example, filter events with a string like this: filter=action:ALLOW AND containerSha:dc58cab81c9a1edb8cd39d34a8a61942c56dc1d4ad27668684be4169d4f0cec5
pageNumber	The page to be returned. Page numbers start with 1.
pageSize	The number of records per page to be included in the response. When not specified you'll get 10 events.

Sample for returning all events with Standard type

You'll get up to 10 events in the output by default.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/events?eventType=
STANDARD' \
--header 'Authorization: Bearer <token>'
```

Response:


```
[
    {
        "customerUuid": "6e0afd12-479c-db0d-822a-793a56bfe353",
        "containerSha":
"3368ab5ebbccb9d17d45cf62f6fa289edade4af81ef5a94e04a4406a1904175d",
        "eventType": "STANDARD",
        "uuid": "70b0dd00-cde7-11ea-8000-a130bd09cb71",
        "dateCreated": 1595620450000,
        "action": "DENY",
        "bindAddress": null,
        "bindPort": 0,
        "fileName": "/etc/passwd",
        "openMode": 0,
        "processId": 42,
        "processName": "/usr/bin/cat",
        "seen": 1,
        "system": "amd64",
        "systemCall": 2,
        "systemCallName": "sys_open"
    },
    {
        "customerUuid": "6e0afd12-479c-db0d-822a-793a56bfe353",
        "containerSha":
"3368ab5ebbccb9d17d45cf62f6fa289edade4af81ef5a94e04a4406a1904175d",
        "eventType": "STANDARD",
        "uuid": "70b0dd00-cde7-11ea-8000-51fe233a28cb",
        "dateCreated": 1595620450000,
        "action": "DENY",
        "bindAddress": null,
        "bindPort": 0,
        "fileName": "/etc/passwd",
        "openMode": 0,
        "processId": 43,
        "processName": "/usr/bin/cat",
        "seen": 1,
        "system": "amd64",
        "systemCall": 2,
        "systemCallName": "sys_open"
    },
...
]
```

More Samples

Try these additional samples in your account.

Sample with Page Number and Page Size specified

In this sample we've specified the number of events to show in the output.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/events?eventType=
STANDARD&pageNumber=1&pageSize=5' \
--header 'Authorization: Bearer <token>'
```

Sample to get events with certain action

In this sample the filter parameter is used to get events with the ALLOW action. Be sure to specify the action value in all caps (ALLOW, DENY, MONITOR).

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/events?eventType=
BEHAVIOR&filter=action:ALLOW' \
--header 'Authorization: Bearer <token>'
```

Sample to get events created within a particular date range

In this sample we'll get events created between June 30, 2020 and July 1, 2020.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/events?eventType=
BEHAVIOR&startTime=2020-06-30T08:30:29&endTime=2020-07-01T08:30:29' \
--header 'Authorization: Bearer <token>'
```

Samples using filter string as input

In this sample we'll only get events for the specified container.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/events?eventType=
BEHAVIOR&filter=containerSha:dc58cab81c9a1edb8cd39d34a8a61942c56dc1d4ad27
668684be4169d4f0cec5' \
--header 'Authorization: Bearer <token>'
```

In this sample we'll only get events with the ALLOW action for the specified container.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/events?eventType=
BEHAVIOR&filter=action:ALLOW AND
containerSha:dc58cab81c9a1edb8cd39d34a8a61942c56dc1d4ad27668684be4169d4f0
cec5' \
--header 'Authorization: Bearer <token>'
```

Policies

Here is the list of the APIs we currently support for policies:

API Objective	Operator	API Path
Get all policies in your account	GET	/csapi/v1.2/runtime/policies
Get details for a specific policy	GET	/csapi/v1.2/runtime/policies/{policyId}
Create a new security policy	POST	/csapi/v1.2/runtime/policies
Update a security policy	PUT	/csapi/v1.2/runtime/policies/{policyId}
Get containers running a policy	GET	/csapi/v1.2/runtime/policies/{policyId}/containers

Samples for various operations on policies:

Get all policies in your account

/csapi/v1.2/runtime/policies

[GET]

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/policies' \
--header 'Authorization: Bearer <token>'
```

Response:

```
[
    {
        "ID": "5e171bef8530d7000151408e",
        "Name": "Default Policy",
        "DateCreated": "2020-01-09T12:26:23.496Z",
        "DateUpdated": "2020-01-09T12:26:23.496Z",
        "Description": "Default group policy",
        "Mode": 0
    },
    {
        "ID": "5e171c738530d70001514091",
        "Name": "Prevent tampering to hosts file",
        "DateCreated": "2020-01-09T12:28:35.761Z",
        "DateUpdated": "2020-01-09T12:28:35.761Z",
        "Description": "Modifications to 'hosts' and 'resolve.conf' file
```

```
can result in resolution of Domain name to malicious IP. This policy
checks for the 'Write' event on either of the specified files",
        "Mode": 0
    },
    {
        "ID": "5e81cf5df12860000129938c",
        "Name": "Deny access in etc v11 Updating With PUT",
        "DateCreated": "0001-01-01T00:00:00Z",
        "DateUpdated": "2020-05-29T04:54:16.432Z",
        "Description": "Deny access in /etc dir for important files",
        "Mode": 1
    },
...
]
```


Get details for a specific policy

/csapi/v1.2/runtime/policies/{policyId}

[GET]

Input Parameters:

Parameter	Description
policyId	(Required) Specify the ID of a specific policy for which you want to get details.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/policies/5eba6fef
2c79c40001e23488' \
--header 'Authorization: Bearer <token>'
```

Response:


```
{
    "ID": "5eba6fef2c79c40001e23488",
    "Name": "Deny access in etc v11",
    "DateCreated": "2020-05-12T09:44:15.315Z",
    "DateUpdated": "2020-05-12T09:44:15.315Z",
    "SchemaVersion": "v1.0",
    "DefaultNetworkAction": "allow",
    "DefaultProgramAction": "allow",
    "DefaultFileAction": "allow",
    "Rules": [
        {
            "ID": "5eba6fef2c79c40001e23489",
            "Name": "Deny access in cat /etc/passwd v01 -- 123789",
            "DateCreated": "0001-01-01T00:00:00Z",
            "DateUpdated": "0001-01-01T00:00:00Z",
            "InActive": false,
```

```
    "RuleType": "syscall",
    "Program": "/bin/cat",
    "Action": "deny",
    "File": "/etc/passwd",
    "ListeningPort": 0,
    "ListeningAddr": "",
    "Protocol": 0,
    "RemotePort": 0,
    "RemoteIps": "",
    "Syscall": "sys_open",
    "SyscallGroup": "",
    "Arg1": "/etc/passwd",
    "Arg2": "",
    "Arg3": ""
},
{
    "ID": "5eba6fef2c79c40001e2348a",
   "Name": "step1_v1.0 - Deny write in etc/hosts file -- 123789",
    "DateCreated": "0001-01-01T00:00:00Z",
    "DateUpdated": "0001-01-01T00:00:00Z",
    "InActive": false,
    "RuleType": "syscall",
    "Program": "/usr/bin/vi",
    "Action": "deny",
    "File": "/etc/hosts",
    "ListeningPort": 0,
    "ListeningAddr": "",
    "Protocol": 0,
    "RemotePort": 0,
    "RemoteIps": "",
    "Syscall": "sys_select",
    "SyscallGroup": "",
    "Arg1": "/etc/hosts",
    "Arg2": "",
    "Arg3": ""
},
{
    "ID": "5eba6fef2c79c40001e2348b",
   "Name": "step2_v1.0 - Deny write in etc/hosts file -- 1237894",
    "DateCreated": "0001-01-01T00:00:00Z",
    "DateUpdated": "0001-01-01T00:00:00Z",
    "InActive": false,
    "RuleType": "syscall",
    "Program": "/usr/bin/vi",
    "Action": "deny",
    "File": "/etc/hosts",
    "ListeningPort": 0,
    "ListeningAddr": "",
    "Protocol": 0,
    "RemotePort": 0,
    "RemoteIps": "",
    "Syscall": "sys_write",
    "SyscallGroup": "",
    "Arg1": "/etc/hosts",
```

```
                "Arg2": "",
                "Arg3": ""
            }
        ],
        "IgnoredSyscalls": [],
        "Mode": 0,
        "Behavioral": true,
        "Description": "Deny access in /etc dir for important files"
    }
```

Create a new security policy

/csapi/v1.2/runtime/policies

[POST]

Input Parameters:

Parameter	Description
Name	Specify a name for the policy.
Description	Provide a description of your policy.
SchemaVersion	Enter the schema version for this policy.
Mode	Specify the policy mode for CRS agents. Enter the numeric value for the mode. Possible values: Active: 0 (the default) Inactive: 1 Permissive: 2
Rules	Policy rules defining control for this policy specified within an array. See Rule Parameters below.

Rule Parameters

Specify rules within an array. These rules will define control for the policy.

Parameter	Description
Name	Specify a name for the rule.
InActive	Specify whether the rule is inactive. Specify false (the default) if the rule is active. Specify true if the rule is not active.
RuleType	Specify the type of rule. Possible values: file, listener, network, syscall.
Program	Specifies path to program that this rule applies to. Wildcards are allowed. The default value is "*".
ListeningPort	Specify the local network port number that this rule applies to. Used only by network and listener rule types.

ListeningAddr	Specify the local IP address that this rule applies to. Used only by network rule types.
Protocol	Specify the network protocol that this rule applies to. Used only by network rules.
RemotePort	Specify the remote network port number that this rule applies to. Used only by network rules.
Syscall	System call name, for rules where an individual system call is to be blocked. Used only in syscall rules.
Arg1	Variable argument. Usage differs depending on rule type.
Arg2	Variable argument. Usage differs depending on rule type.
Arg3	Variable argument. Usage differs depending on rule type.
File	Specify the path to the file that the rule applies to.
Action	Specify the action that should be taken if this rule is matched. Possible values: allow, deny, monitor
DateCreated	Timestamp for when object was created in the format ['YYYY'-'MM'-'DD'T'hh':'mm':'ss'.'sss'Z].
DateUpdated	Timestamp for when object was last updated in the format ['YYYY'-'MM'-'DD'T'hh':'mm':'ss'.'sss'Z].
RemoteIps	Specify IPs restricted in network rules.

API request:

```
curl --location --request POST
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/policies' \
--header 'Authorization: Bearer <token>'
--header 'Content-Type: text/plain' \
--data-raw '{
    "Name": "Monitor /etc/hosts File Access ",
    "SchemaVersion": "v1.0",
    "Rules": [
        {
            "Name": "Modification_hosts_Monitor",
            "DateCreated": "2019-11-04T00:00:00Z",
            "DateUpdated": "2019-11-04T00:00:00Z",
            "InActive": false,
            "RuleType": "syscall",
            "Program": "*",
            "Action": "monitor",
            "File": "/etc/hosts",
            "Syscall": "sys_write",
            "Arg1": "/etc/hosts",
            "Arg2": "",
            "Arg3": ""
        },
```

```
            {
                "Name": "Modification_resolvconf_deny",
                "DateCreated": "2019-11-04T00:00:00Z",
                "DateUpdated": "2019-11-04T00:00:00Z",
                "InActive": false,
                "RuleType": "syscall",
                "Program": "*",
                "Action": "deny",
                "File": "/etc/resolv.conf",
                "Syscall": "sys_write",
                "Arg1": "/etc/resolv.conf",
                "Arg2": "",
                "Arg3": ""
            }
        ],
        "Mode": 0,
        "Description": "Modifications to 'hosts' and 'resolve.conf' file can
    result in resolution of Domain name to malicious IP. This policy checks
    for the 'Write' event on either of the specified files"
    }
```

Response:

```
    {
        "ID": "5ede4881ca13c90001f86e8f",
        "Name": "Monitor /etc/hosts File Access",
        "DateCreated": "2020-06-08T14:17:37.626Z",
        "DateUpdated": "2020-06-08T14:17:37.626Z",
        "SchemaVersion": "v1.0",
        "DefaultNetworkAction": "allow",
        "DefaultProgramAction": "allow",
        "DefaultFileAction": "allow",
        "Rules": [
            {
                "ID": "5ede4881ca13c90001f86e90",
                "Name": "Modification_hosts_Monitor",
                "DateCreated": "2019-11-04T00:00:00Z",
                "DateUpdated": "2019-11-04T00:00:00Z",
                "InActive": false,
                "RuleType": "syscall",
                "Program": "*",
                "Action": "monitor",
                "File": "/etc/hosts",
                "ListeningPort": 0,
                "ListeningAddr": "",
                "Protocol": 0,
                "RemotePort": 0,
                "RemoteIps": "",
                "Syscall": "sys_write",
                "SyscallGroup": "",
                "Arg1": "/etc/hosts",
                "Arg2": "",
                "Arg3": ""
            },
```

```
        {
            "ID": "5ede4881ca13c90001f86e91",
            "Name": "Modification_resolvconf_deny",
            "DateCreated": "2019-11-04T00:00:00Z",
            "DateUpdated": "2019-11-04T00:00:00Z",
            "InActive": false,
            "RuleType": "syscall",
            "Program": "*",
            "Action": "deny",
            "File": "/etc/resolv.conf",
            "ListeningPort": 0,
            "ListeningAddr": "",
            "Protocol": 0,
            "RemotePort": 0,
            "RemoteIps": "",
            "Syscall": "sys_write",
            "SyscallGroup": "",
            "Arg1": "/etc/resolv.conf",
            "Arg2": "",
            "Arg3": ""
        }
    ],
    "IgnoredSyscalls": [],
    "Mode": 0,
    "Behavioral": true,
    "Description": "Modifications to 'hosts' and 'resolve.conf' file can
result in resolution of Domain name to malicious IP. This policy checks
for the 'Write' event on either of the specified files"
}
```

Update a security policy

/csapi/v1.2/runtime/policies/{policyId}

[PUT]

Input Parameters:

Parameter	Description
policyId	(Required) Specify the ID of the policy to update.
Name={value}	Specify a name for the policy.
Description={value}	Provide a description of your policy.
SchemaVersion={value}	Enter the schema version for this policy.
Mode={value}	Specify the policy mode for CRS agents. Enter the numeric value for the mode. Possible values: Active: 0 (the default) Inactive: 1 Permissive: 2
Rules={value}	Policy rules defining control for this policy specified within an array. See Rule Parameters in previous section.

API request:

```
curl --location --request PUT
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/policies/5ede4881
ca13c90001f86e8f' \
--header 'Authorization: Bearer <token>'
--header 'Content-Type: text/plain' \
--data-raw '{{
    "Name": "Deny /etc/hosts File Access",
    "SchemaVersion": "v1.0",
    "Rules": [
        {
            "Name": "Modification_hosts_Monitor",
            "DateCreated": "2019-11-04T00:00:00Z",
            "DateUpdated": "2019-11-04T00:00:00Z",
            "InActive": false,
            "RuleType": "syscall",
            "Program": "*",
            "Action": "deny",
            "File": "/etc/hosts",
            "Syscall": "sys_write",
            "Arg1": "/etc/hosts",
            "Arg2": "",
            "Arg3": ""
        },
        {
            "Name": "Modification_resolvconf_deny",
            "DateCreated": "2019-11-04T00:00:00Z",
```

```
                "DateUpdated": "2019-11-04T00:00:00Z",
                "InActive": false,
                "RuleType": "syscall",
                "Program": "*",
                "Action": "deny",
                "File": "/etc/resolv.conf",
                "Syscall": "sys_write",
                "Arg1": "/etc/resolv.conf",
                "Arg2": "",
                "Arg3": ""
            }
        ],
        "Mode": 0,
        "Description": "Modifications to 'hosts' and 'resolve.conf' file can
    result in resolution of Domain name to malicious IP. This policy checks
    for the 'Write' event on either of the specified files"
    }
```

Response:

```
    {
        "ID": "5ede4881ca13c90001f86e8f",
        "Name": "Deny /etc/hosts File Access ",
        "DateCreated": "0001-01-01T00:00:00Z",
        "DateUpdated": "2020-06-09T04:32:06.108Z",
        "SchemaVersion": "v1.0",
        "DefaultNetworkAction": "allow",
        "DefaultProgramAction": "allow",
        "DefaultFileAction": "allow",
        "Rules": [
            {
                "ID": "5edf10c64b23720001a7556a",
                "Name": "Modification_hosts_Monitor",
                "DateCreated": "2019-11-04T00:00:00Z",
                "DateUpdated": "2019-11-04T00:00:00Z",
                "InActive": false,
                "RuleType": "syscall",
                "Program": "*",
                "Action": "deny",
                "File": "/etc/hosts",
                "ListeningPort": 0,
                "ListeningAddr": "",
                "Protocol": 0,
                "RemotePort": 0,
                "RemoteIps": "",
                "Syscall": "sys_write",
                "SyscallGroup": "",
                "Arg1": "/etc/hosts",
                "Arg2": "",
                "Arg3": ""
            },
            {
                "ID": "5edf10c64b23720001a7556b",
                "Name": "Modification_resolvconf_deny",
```

```
            "DateCreated": "2019-11-04T00:00:00Z",
            "DateUpdated": "2019-11-04T00:00:00Z",
            "InActive": false,
            "RuleType": "syscall",
            "Program": "*",
            "Action": "deny",
            "File": "/etc/resolv.conf",
            "ListeningPort": 0,
            "ListeningAddr": "",
            "Protocol": 0,
            "RemotePort": 0,
            "RemoteIps": "",
            "Syscall": "sys_write",
            "SyscallGroup": "",
            "Arg1": "/etc/resolv.conf",
            "Arg2": "",
            "Arg3": ""
        }
    ],
    "IgnoredSyscalls": [],
    "Mode": 0,
    "Behavioral": true,
    "Description": "Modifications to 'hosts' and 'resolve.conf' file can
result in resolution of Domain name to malicious IP. This policy checks
for the 'Write' event on either of the specified files"
}
```

Get containers running a specific policy

/csapi/v1.2/runtime/policies/{policyId}/containers

[GET]

Input Parameters:

Parameter	Description
policyId	(Required) Specify the ID of a specific policy for which you want to get a list of containers running the policy.

API request:

```
curl --location --request GET
'https://gateway.qg1.apps.qualys.com/csapi/v1.2/runtime/policies/5eddddaf
4b23720001a7552f/containers' \
--header 'Authorization: Bearer <token>'
```

Response:


```
[
    {
        "ContainerSHA":
"e6a1944ffc0dcb07de39b857a2b1f314d34be6f68b2f2d3b9a406a3a52bc17ed",
        "ConfigID": "5eccc9ca906942000123c111",
        "Status": "stopped"
    },
    {
        "ContainerSHA":
"3bcf2d070db4601cfdb850b912c3800e34ade1a2130728f725dd8e7357f9f9c1",
        "ConfigID": "5eccc9ca906942000123c111",
        "Status": "stopped"
    },
    {
        "ContainerSHA":
"4eaf6b2806f0402d78dc77ceac805a74c20ee75efb520dbceb7300f18d90b6b0",
        "ConfigID": "5eccc9ca906942000123c111",
        "Status": "stopped"
    }
]
```