Tenable.io Container Security REST API

Last Revised: June 08, 2017

Tenable.io Container Security API

Tenable.io Container Security includes a number of APIs for interacting with the platform:

- Reports API
- Test Jobs API
- Policy API
- Containers API

Caution: To utilize the Tenable.io Container Security API, requests require authentication using Access and Secret keys sent with their headers.

Reports API

The Reports API is used to obtain Container Security reports in either JSON or Nessus V2 file format.

- JSON Report by Container ID
- JSON Report by Docker Image ID
- JSON Report by Image Digest
- Nessus Report by Container Image ID

JSON Report by Container ID

API endpoint: /api/v1/reports/show

This call returns a report in JSON format for a container that you specify by ID.

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-
security/api/v1/reports/show?container_id={container_id}
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key
secretKey	Your Tenable.io API Secret key
container_id	the ID of the container image

Note: If you do not have the container_id, you can call the List Stored Container Images endpoint.

JSON Report by Docker Image ID

API endpoint: /api/v1/reports/by_image

This call returns a report in JSON format for an image that you specify by Docker ID:

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-security/api/v1/reports/by_
image?image_id={image_id}
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key
secretKey	Your Tenable.io API Secret key
image_id	the ID of the image as returned by docker

JSON Report by Image Digest

API endpoint: /api/v1/reports/by_image_digest

This call returns a report in JSON format for an image that you specify using the SHA256 hash of the image.:

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-security/api/v1/reports/by_image_
digest?image_digest={sha256}
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key
secretKey	Your Tenable.io API Secret key
sha256	SHA256 checksum of the container image

Note: This is helpful in cases where you have previously submitted the container image to Tenable.io Container Security and are seeking to pull the same (or a newer copy) of the test result again.

Nessus Report by Container Image ID

API endpoint: /api/v1/reports/nessus/show

This call returns a Tenable.io Container Security report prepared in Nessus V2 file format. You can then import the report into Nessus Professional (versions 4.0 and higher):

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-
security/api/v1/reports/nessus/show?id={id}
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key
secretKey	Your Tenable.io API Secret key
id	the ID of the container image

Note: If you do not have the container_id, you may call the List Stored Container Images endpoint.

Test Jobs API

The Test Jobs API is used to query the status of container tests. You can get a list of all current and recent tests, or request the status of tests for specific images.

- Get Status of Test Job
- Job Status by Docker Image ID
- Get Job Status by Docker Image Digest
- List All Test Jobs

Get Job Status

API endpoint: /api/v1/jobs/status

Request the status of a test to determine if a test is stilled queued, in progress, or has completed:

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-security/api/v1/jobs/status?job_id=
{job_id}
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key
secretKey	Your Tenable.io API Secret key
job_id	the job ID of the queued job

Note: If you do not have the job_id, you may call the List All Test Jobs endpoint.

Get Job Status by Docker Image ID

API endpoint: /api/v1/jobs/image_status

Request the status of a test by the Docker Image ID to determine if a test is stilled queued, in progress, or has completed:

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-security/api/v1/jobs/image_
status?image_id={image_id}
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key
secretKey	Your Tenable.io API Secret key
image_id	the ID of the image as returned by docker

Get Job Status by Docker Image Digest

API endpoint: /api/v1/jobs/image_status_digest

Request the status of a job by the Docker Image ID to determine if a test is stilled queued, in progress, or has completed:

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-security/api/v1/jobs/image_status_
digest?image_digest={sha256}
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key

Parameter	Description
secretKey	Your Tenable.io API Secret key
sha256	SHA256 checksum of the container image

List All Test Jobs

API endpoint: /api/v1/jobs/list

Requests a list of active and recent jobs:

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-security/api/v1/jobs/list
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key
secretKey	Your Tenable.io API Secret key

Policy API

Query for the compliance status of images.

- Policy Compliance Status by Docker Image ID

Policy Compliance Status by Docker Image ID

API endpoint: /api/v1/policycompliance

To query for the policy compliance status of an image, query by the Docker Image ID and authenticate using Access and Secret keys:

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-
security/api/v1/policycompliance?image_id={image_id}
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key
secretKey	Your Tenable.io API Secret key
image_id	the ID of the image as returned by docker

Container Images API

Manage container images that have been pushed to Tenable.io Container Security.

- List Stored Container Images
- Delete a Container Image from the Repository

List Stored Container Images

API endpoint: /api/v1/container/list

Tenable.io Container Security provides an API endpoint for enumerating container images stored by a given user, by Access and Secret keys:

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-security/api/v1/container/list
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key
secretKey	Your Tenable.io API Secret key

Note: The list stored container images call returns container_image_id's owned by the user matching the provided Access and Secret keys.

Delete a Container Image from the Repository

API endpoint: /api/v1/container/{repositoryName}/manifests/{sha256}

Delete a container image by providing the registry and path for the repository, as well as the SHA256 hash of the image:

```
$ curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}" -X
DELETE https://cloud.tenable.com/container-security/api/v1/container/
{repositoryName}/manifests/{sha256}
```

Query Parameters

Parameter	Description
accessKey	Your Tenable.io API Access key
secretKey	Your Tenable.io API Secret key
repositoryName	The registry/path of the image. For example, if you have a repository named *unix* and an image named *alpine*, you would type *unix/alpine*.
sha256	SHA256 checksum of the container image

Authorizing your app to use the Tenable.io Container Security API

To utilize the Tenable.io Container Security API, requests require authentication using Access and Secret keys sent with their headers.

API Keys

If desired, you can generate Access and Secret keys via the Tenable.io interface. Alternatively, these keys can be generated per account through session: keys or users: keys and can be used to authenticate without creating a session.

Add them to your request using the following HTTP header:

```
X-ApiKeys: accessKey={accessKey}; secretKey={secretKey};
```

Example:

```
curl -H "X-ApiKeys: accessKey={accessKey}; secretKey={secretKey}"
https://cloud.tenable.com/container-security/api/v1/jobs/status?job_id=-
=<job_id>
```

In addition, it is possible to perform actions as if authenticated as a different user by adding an additional HTTP header:

```
X-Impersonate: username={username}
```

Example:

```
curl -H "X-Impersonate: username={username}" -H "X-ApiKeys: accessKey=
{accessKey}; secretKey={secretKey}" https://cloud.tenable.com/container-
security/api/v1/jobs/status?job_id=<job_id>
```

User Roles

Name	Value	Description
Basic	16	Users with this role can view and configure scan results.
Standard	32	Users with this role can create scans, policies, and user target groups.
Administrator	64	Users with this role have the same privileges as the standard user but can also manage users, groups, agents, exclusions, target groups, and scanners.

session: keys

Generates API Keys for the current user.

Request

This request requires basic user permissions.

HTTP Request

```
PUT /session/keys
```

Parameters

This request does not have any parameters.

Response

Status Code	Description
200	Returned if the user API Keys were generated.
401	Returned if the user is not logged in.

If successful, this method returns a response body with the following structure:

```
{
    "accessKey": {string},
    "secretKey": {string}
}
```

users: keys

Generates the API Keys for the given user.

Request

This request requires administrator user permissions.

HTTP Request

```
PUT /users/{user_id}/keys
```

Parameters

Parameter	Value	Description
user_id	integer	The unique id of the user.

Response

Status Code	Description
200	Returned if the user API Keys were generated.
403	Returned if the user does not have permission to generate API keys.
404	Returned if the user does not exist.
500	Returned if the server failed to change the keys.

If successful, this method returns a response body with the following structure:

```
{
    "accessKey": {string},
    "secretKey": {string}
}
```