Juniper Networks | Engineering Simplicity

Juniper ATP Cloud Threat Intelligence Open API Setup Guide

Published
2023-07-25

Table of Contents

Threat Intelligence Open API

Threat Intelligence Open API

IN THIS SECTION

- Threat Intelligence API Overview | 1
- Juniper ATP Cloud API Overview | 4
- File/Hash API Overview | 7
- Infected Host API Overview | 8
- IP Filter API Overview | 8
- Example | 9
- SRX Series Update Intervals for Cloud Feeds | 11
- Open API for DNS Category | 12

Juniper Advanced Threat Prevention Cloud (Juniper ATP Cloud) provides the following APIs that can help you keep your network free of sophisticated malware and cyberattacks by using superior cloud-based protection:

Threat Intelligence API Overview

The Threat Intelligence open API allows you to program the Juniper ATP Cloud Command and Control server (C&C) feeds to suit your requirements. You can perform the following operations using the threat intelligence API:

- Inject an IP, URL, or domain into a C&C feed with a threat level from 1 through 10. You can create up to 30 different custom C&C feeds.
  - An IP can be an IP address, IP range, or IP subnet.
  - Both IPv4 and IPv6 addresses are supported.
- Update the threat level of an IP, URL, or domain from 1 through 10.
- Delete a specific server in the feed or delete the entire feed.
- Retrieve the current status of an operation (processing) or errors (if any) from the feed processing engine.

The Threat Intelligence API supports a Swagger API specification in JSON format to allow programmatic access to it. For more information on the Swagger API specification, see https://threat-api.sky.junipersecurity.net/swagger.json.

NOTE: Starting in Junos OS 19.2, SRX Series Firewalls support inspection of encrypted traffic (HTTPS) as well as HTTP traffic in threat intelligence feeds. Server name identification (SNI) checks are also supported. These changes do not introduce new CLI commands. Existing commands make use of this functionality.

The following table lists the rate limits (number of requests you can make per minute) for the Threat Intelligence APIs. If you exceed these rate limits, you will receive a `429 - Too many Requests` error.

Feature	Maximum Number of Requests Per Minute
C&C feed	60
Blocklist feed	60
Allowlist feed	60
DNS	50

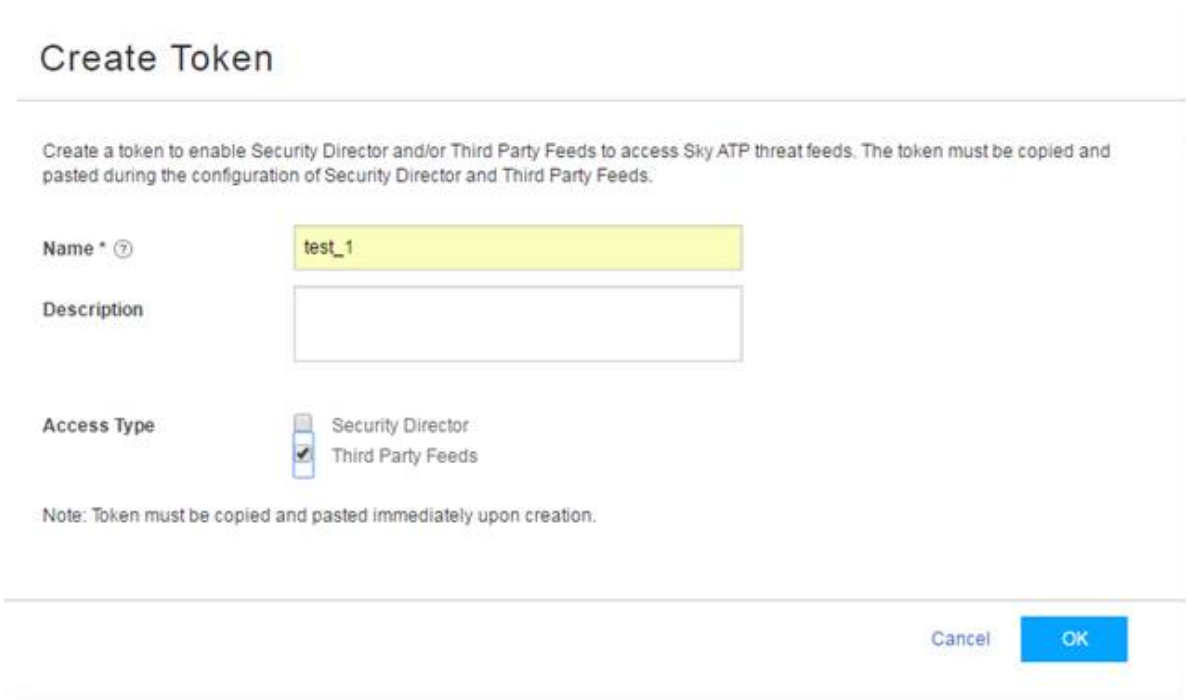
Configuration and Setup

To access the API, you must create an application token in the Juniper ATP Cloud Web UI and use that token as the bearer token in the authorization header.

To generate an application token:

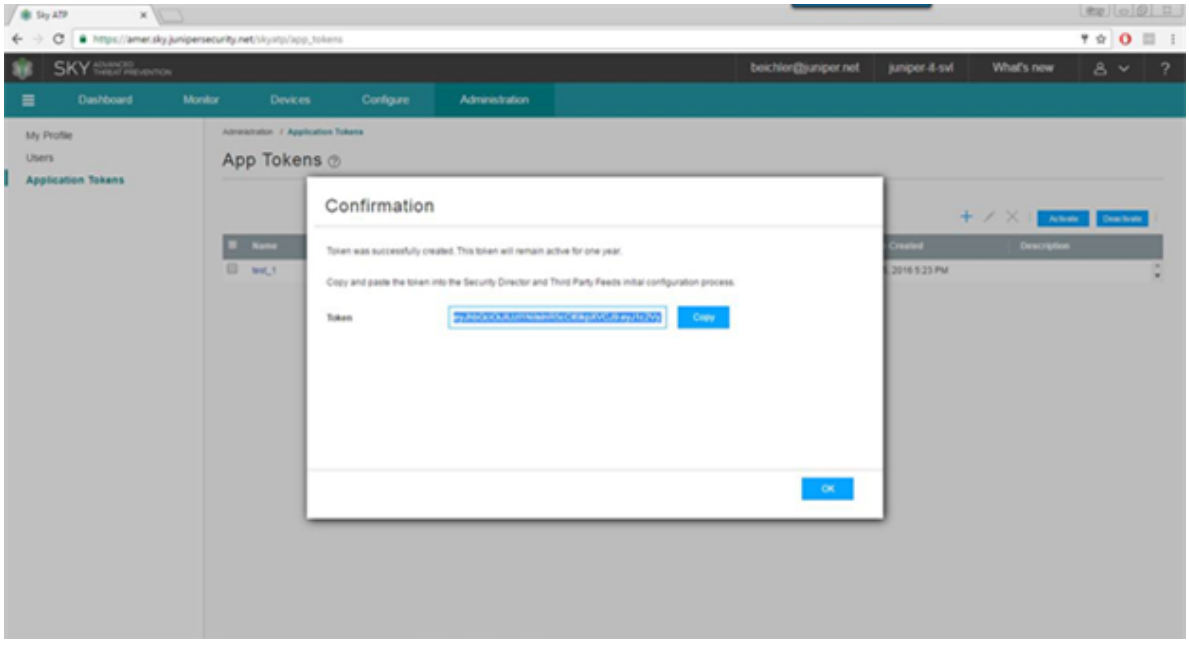
1. Log in to the Juniper ATP Cloud Web UI using your credentials. Select **Administration > Application Tokens** and click the plus (+) sign. Fill in the name of the token and other required details in the pop-up box that appears and click **OK** to create a new token. See Figure 1 on page 3.

Figure 1: Creating an Application Token



2. A confirmation pop-up message appears, indicating the creation of a new token, as shown in Figure 2 on page 3. You can now use this token to access the Juniper ATP Cloud API.

Figure 2: Confirming the Creation of an Application Token



> **NOTE**: You can generate a maximum of 10 tokens per user, and each token is valid for one year.

For more information on how to create application tokens, see Creating Application Tokens.

Usage Examples

The following cURL examples illustrate the use of the threat intelligence API:

- `curl -k -v -XPOST -H "Authorization: Bearer <TOKEN>" -F file=@/tmp/whitelist.txt <API HOST>/v1/cloudfeeds/whitelist/file/ip/<FEEDNAME>`
- `curl -k -v -XPOST -H "Authorization: Bearer <TOKEN>" -F file=@/tmp/whitelist.txt <API HOST>/v1/cloudfeeds/cc/file/ip/<FEEDNAME>`

where:

- `API HOST` is the name of the Open API hostname corresponding to the location of the customer portal. Please refer to Table 1 on page 5 for the correct hostname for your location.
- *TOKEN* is the application token generated in the Juniper ATP Cloud Web UI.
- *FEED NAME* is the name of the feed you want to create.

Juniper ATP Cloud API Overview

You can perform the following operations using the Juniper ATP Cloud API:

- Retrieve the blocklist or allowlist for the specific server type.
- Update an IP, URL, or FQDN in a blocklist or allowlist server list.
  - An IP can be an IP address, IP range, or IP subnet.
  - Both IPv4 and IPv6 addresses are supported.
- Delete a specific server in the list or delete the entire list.

The Juniper ATP Cloud API supports a Swagger API specification in JSON format to allow programmatic access to it. For more information on the Swagger API specification, see https://api.sky.junipersecurity.net/swagger.json.

The following table lists the rate limits (number of requests you can make per minute) for the Juniper ATP Cloud APIs. If you exceed these rate limits, you will receive a `429 - Too many Requests` error.

Feature	Maximum Number of Requests Per Minute
Hash lookup	50
File submissions	10
Blocklist	60
Allowlist	60

NOTE: Juniper ATP Cloud supports up to 3,000 entries in the allowlist and 3,000 entires in the blocklist.

Configuration and Setup

To access the API, you must create an application token in the Juniper ATP Cloud Web UI and use that token as the bearer token in the authorization header. See section "Configuration and Setup" on page 2 for more information on the creation of the token.

Juniper ATP Cloud URLs

Juniper ATP Cloud hostnames varies by location. Please refer to the following table:

Table 1: Juniper ATP Cloud URLs by Location

Location	Juniper ATP Cloud URL
United States	Customer Portal: **https://amer.sky.junipersecurity.net** Open API (infected hosts, allowlist/blocklist, sample submission): **https://api.sky.junipersecurity.net** Open API (threat intelligence): **https://threat-api.sky.junipersecurity.net**

Table 1: Juniper ATP Cloud URLs by Location *(Continued)*

Location	Juniper ATP Cloud URL
European Union	Customer Portal: **https://euapac.sky.junipersecurity.net** Open API (infected hosts, allowlist/blocklist, sample submission): **https://api-eu.sky.junipersecurity.net** Open API (threat intelligence): **https://threat-api.sky.junipersecurity.net**
APAC	Customer Portal: **https://apac.sky.junipersecurity.net** Open API (infected hosts, allowlist/blocklist, sample submission): **https://api-apac.sky.junipersecurity.net** Open API (threat intelligence): **https://threat-api-apac.sky.junipersecurity.net**
Canada	Customer Portal: **https://canada.sky.junipersecurity.net** Open API (infected hosts, allowlist/blocklist, sample submission): **https://api-canada.sky.junipersecurity.net** Open API (threat intelligence): **https://threat-api-canada.sky.junipersecurity.net**

Usage Example

The following cURL example illustrates the use of the Juniper ATP Cloud API:

- `curl -k -v -XPOST -H "Authorization: Bearer <TOKEN>" -F file=@/tmp/blacklist.txt <API HOSTNAME>/v1/skyatp/blacklist/file/ip/<FEED NAME>`

where:

- `API HOST` is the name of the Open API hostname corresponding to the location of the customer portal. Please refer to Table 1 on page 5 for the correct hostname for your location.
- *`TOKEN`* is the application token generated in the Juniper ATP Cloud Web UI.
- *`FEED NAME`* is the name of the feed you want to create.

File/Hash API Overview

The file/hash API lets you submit files for analysis. You can perform the following operations:

- Look up sample malware scores by hash.
- Submit samples for malware analysis.
- Update an IP, URL, or FQDN from a file in a specific list.
  - An IP can be an IP address, IP range, or IP subnet.
  - Both IPv4 and IPv6 addresses are supported.

The file/hash API supports a Swagger API specification in JSON format to allow programmatic access to it. For more information on the Swagger API specification, see https://api.sky.junipersecurity.net/swagger.json.

Configuration and Setup

To access the API, you must create an application token in the Juniper ATP Cloud Web UI and use that token as the bearer token in the authorization header. See section "Configuration and Setup" on page 2 for more information on the creation of the token.

Usage Example

The following cURL example illustrates the use of the file/hash API:

- `curl -H "Authorization: Bearer <TOKEN>" -k <API HOSTNAME>/v1/skyatp/lookup/hash/<SHA256>?full_report=true`
- `curl -H "Authorization: Bearer <TOKEN>" -k -F file=@/srv/sample.exe <API HOSTNAME>/v1/skyatp/submit/sample`

> **NOTE**: `API HOST` is the name of the Open API hostname corresponding to the location of the customer portal. Please refer to Table 1 on page 5 for the correct hostname for your location.

where:

- *TOKEN* is the application token generated in the Juniper ATP Cloud Web UI.
- *SHA256* is the sample hash. Only SHA256 is supported at this time.

Full reports will be completely supported in an upcoming release. The report you receive right now may slightly different in appearance and content.

Infected Host API Overview

The infected host feed is generated by Juniper ATP Cloud and is used to flag compromised hosts. The feed is dynamic. Hosts are automatically added when Juniper ATP Cloud suspects a host has been compromised (through a proprietary algorithm) and can be manually removed from the list through the user interface once you feel the host is no longer compromised. The feed lists the IP address or IP subnet of the host along with a threat level, for example, xxx.xxx.xxx.133 and threat level 5. This feed is unique to a realm and IP addresses within the real are assumed to be non-overlapping.

Associated with the infected host feed are a allowlist and blocklist. These are different from the generic Juniper ATP Cloud allowlist and blocklist. The infected host feed uses these lists to remove hosts that are currently on an infected host feed (allowlist) and to always list a host in the infected host feed (blocklist.)

With the infected host API, you can do the following:

- Return a list of all IP addresses in the current infected host feed.
- Return a list of all IP addresses in the infected host allowlist or blocklist.
- Delete an IP address from the infected host allowlist or blocklist.
- Add an IP address to the infected host allowlist or blocklist.

The infected host API supports a Swagger API specification in JSON format to allow programmatic access to it. For more information on the Swagger API specification, see https://api.sky.junipersecurity.net/swagger.json.

Configuration and Setup

To access the API, you must create an application token in the Juniper ATP Cloud Web UI and use that token as the bearer token in the authorization header. See section "Configuration and Setup" on page 2 for more information on the creation of the token.

IP Filter API Overview

A Dynamic Address Entry (DAE) provides dynamic IP address information to security policies. A DAE is a group of IP addresses, not just a single IP prefix, that can be imported. These IP addresses are for specific domains or for entities that have a common attribute such as a particular undesired location that poses a threat. The administrator can then configure security policies to use the DAE within a security policy. When the DAE is updated, the changes automatically become part of the security policy. There is

no need to update the policy manually. Note that this is an IP address-only feed. It does not support URLs or fully qualified domain names (FQDNs).

The IP filter APIs let you perform the following tasks:

- Remove IP addresses (in a .csv file) from an IP filter feed
- Add IP addresses (in a .csv file) to an IP filter feed.
- Remove a specific IP address from the IP filter feed.
- Add a specific IP address to the IP filter feed.
- Remove a specific IP filter feed.
- Get the processing status of a specific IP Filter feed.

The IP filter API supports a Swagger API specification in JSON format to allow programmatic access to it. For more information on the Swagger API specification, see https://api.sky.junipersecurity.net/swagger.json.

Configuration and Setup

To access the API, you must create an application token in the Juniper ATP Cloud Web UI and use that token as the bearer token in the authorization header. See section "Configuration and Setup" on page 2 for more information on the creation of the token.

Example

In this example, targeted attacked are being performed against web servers in a DMZ while concealing their identities via Tor. Tor exit nodes move frequently and keeping an up-to-date list of all 1000+ exit nodes within a firewall policy is almost impossible. This can, however, be done easily using Juniper ATP Cloud APIs. For more information on this example, see Automating Cyber Threat Intelligence with Sky ATP.

Shown below is an example script that performs the following actions:

- Polls the official TorProject's exit-node list via cURL and extracts legitimate IP information via `grep`.
- Utilizes Juniper ATP Cloud open API to install and propagate third-party threat intelligence to all SRX Series Firewalls in the network.

- Runs on an hourly basis via cron to ensure that the active Tor Relays are always being blocked.

```
#!/bin/bash

# Define Application Token (Paste in your value between the "")
APPToken="Your_Application_Token_Here"

# Define the name of the feed you wish to create
FeedName="Tor_Exit_Nodes"

#Define temporary file to store address list
TorList=/var/tmp/torlist.txt

# cURL fetches Tor Relay list from https://check.torproject.org/exit-addresses
# grep identifies and extracts valid IP addresses from the list

curl -k https://check.torproject.org/exit-addresses | grep -E -o
'(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?
[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]
|2[0-4][0-9]|[01]?[0-9][0-9]?)' > $TorList

#Remove old Feed information before uploading new list
curl -k -v -XDELETE -H "Authorization: Bearer $APPToken" -F server='*' https://threat-
api.sky.junipersecurity.net/v1/cloudfeeds/blacklist/param/ip/${FeedName}

# Wait for 5 seconds before uploading new list
sleep 5

#Upload List to SkyATP as Feed Tor_Exit_Nodes
curl -k -v -XPOST -H "Authorization: Bearer $APPToken" -F file=@${TorList} https://threat-
api.sky.junipersecurity.net/v1/cloudfeeds/blacklist/file/ip/${FeedName}

# Cleanup
rm $TorList

# Exit
```

Once the script has been run successfully, we can see that the latest Tor Nodes are being blocked during an ICMP request below (`feed-name=Tor_Exit_Nodes`)

```
<14>1 2016-10-17T15:18:11.618Z SRX-1500 RT_SECINTEL - SECINTEL_ACTION_LOG [junos@x.x.x.x.x.137
category="secintel" sub-category="Blacklist" action="BLOCK" action-detail="DROP" http-host="N/A"
threat-severity="0" source-address="5.196.121.161" source-port="1" destination-
address="x.x.0.10" destination-port="24039" protocol-id="1" application="N/A" nested-
application="N/A" feed-name="Tor_Exit_Nodes" policy-name="cc_policy" profile-name="Blacklist"
username="N/A" roles="N/A" session-id-32="572564" source-zone-name="Outside" destination-zone-
name="DMZ"] category=secintel sub-category=Blacklist action=BLOCK action-detail=DROP http-
host=N/A threat-severity=0 source-address=x.x.0.110 source-port=1 destination-address=x.x.x.161
destination-port=24039 protocol-id=1 application=N/A nested-application=N/A feed-
name=Tor_Exit_Nodes policy-name=cc_policy profile-name=Blacklist username=N/A roles=N/A session-
id-32=572564 source-zone-name=Outside destination-zone-name=DMZ
```

SRX Series Update Intervals for Cloud Feeds

The following table provides the update intervals for each feed type. Note that when the SRX Series Firewall makes requests for new and updated feed content, if there is no new content, no updates are downloaded at that time.

Table 2: Feed Update Intervals

Category	Feeds	SRX Update Intervals (in seconds)
Command and Control	Juniper Feeds	1,800
	Integrated Feeds	1,800
	Customer Feeds	1,800
GeoIP	geoip_country	435,600
Allowlist	Customer Feeds	3,600
Blocklist	Customer Feeds	3,600

Table 2: Feed Update Intervals *(Continued)*

Category	Feeds	SRX Update Intervals (in seconds)
Infected Hosts	Infected Hosts	60
IPFilter	Customer Feeds	1,800
	Office 365	1,800
DNS	DNS Feeds	1,800

Open API for DNS Category

The following table provides the feed manifest that is downloaded by the SRX Series Firewall.

Table 3: Feed Manifest

HTTP Method	URI	Request Body	Response

POST/PATCH	/DNS/file/{feed_name}	File : file.txt
		cat file_content>
		1.1.1.1,10
		1.1.1.2,10
			{
			"request_id": "string"
			}

GET	/DNS/file/{feed_name}		{
			“message”:”string”,
			"request_id": "string"
			}

Table 3: Feed Manifest *(Continued)*

HTTP Method	URI	Request Body	Response

DELETE	/DNS/file/{feed_name}	File : file.txt
		cat file_content>
		1.1.1.1,10
			{
			"request_id": "string"
			}

RELATED DOCUMENTATION

- Threat Intelligence Open API Reference Guide
- Juniper Sky ATP Open API Reference Guide