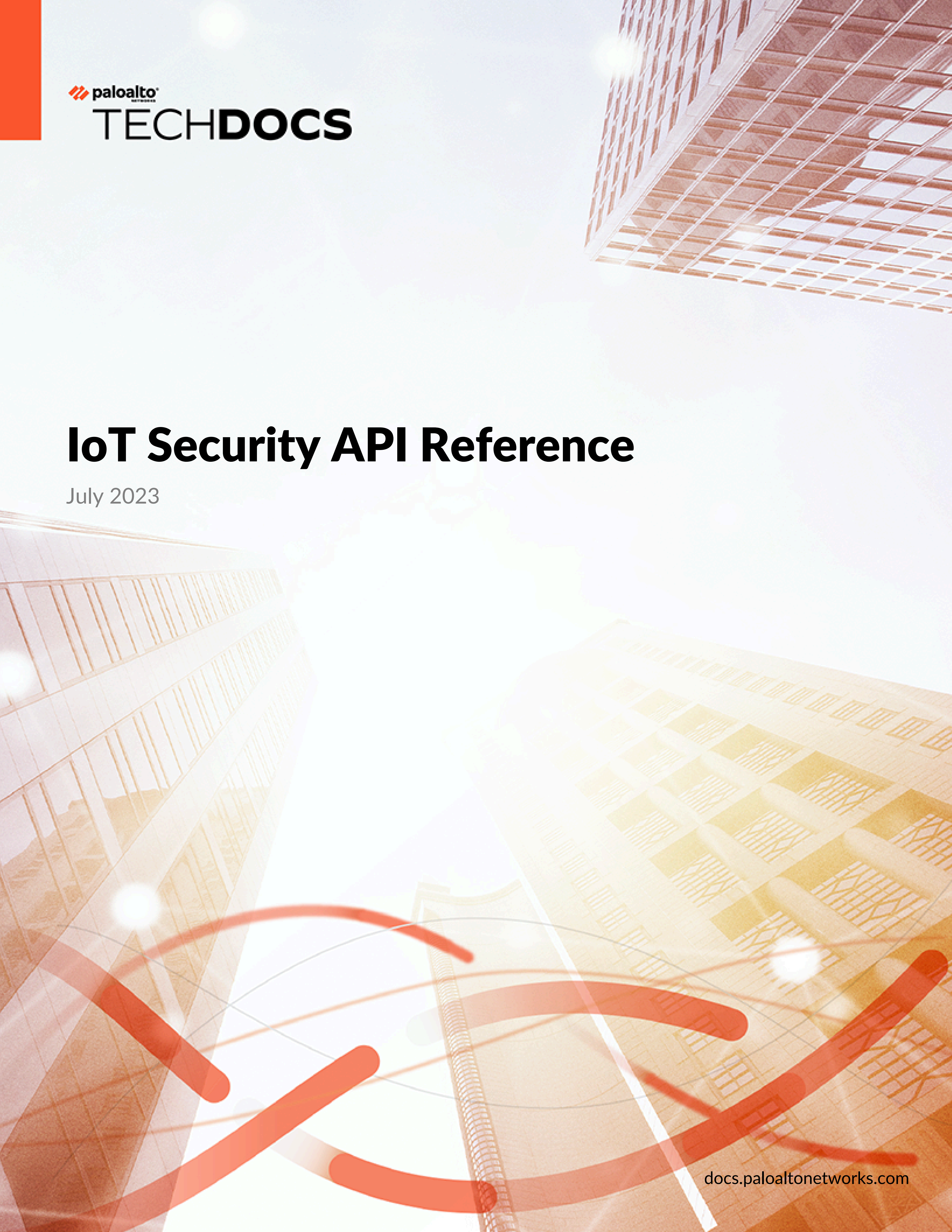


IoT Security API Reference

July 2023

docs.paloaltonetworks.com

Contact Information

Corporate Headquarters:
Palo Alto Networks
3000 Tannery Way
Santa Clara, CA 95054
www.paloaltonetworks.com/company/contact-support

About the Documentation

- For the most recent version of this guide or for access to related documentation, visit the Technical Documentation portal docs.paloaltonetworks.com.
- To search for a specific topic, go to our search page docs.paloaltonetworks.com/search.html.
- Have feedback or questions for us? Leave a comment on any page in the portal, or write to us at documentation@paloaltonetworks.com.

Copyright

Palo Alto Networks, Inc.
www.paloaltonetworks.com

Last Revised

July 11, 2023

Table of Contents

IoT Security API Overview

The IoT Security API lets you integrate IoT Security with third-party apps or services to ingest IoT device inventories, device details, security alerts, and device vulnerabilities. It also lets you resolve alerts and vulnerabilities and add and remove user-defined tags.

- Get Started with the IoT Security API

Get Started with the IoT Security API

The following parameters are used in queries sent to the IoT Security API.

Parameter	Example
Protocol	`https`
Tenant-specific URIs	`acmecorp.iot.paloaltonetworks.com` where `acmecorp` is the tenant ID (customer ID) *A tenant is the organization that owns an IoT Security account.*
Path	`/pub/v4.0/`
Function call	`device` retrieves details about an individual device by device ID. This is typically its MAC address but when a device is configured as a static IP device, the device ID is its IP address. `device/ip` retrieves details about one or more devices by IP address. `device/list` retrieves the entire device inventory for a tenant. `profile/mapping` retrieves a complete list of profile-category-vertical mappings. `alert/list` retrieves the entire list of security and system alerts. `vulnerability/list` retrieves the entire list of vulnerability instances. `alert/update` resolves a security alert. `vulnerability/update` resolves one or more vulnerability instances. `device/update` adds a user tag to one or more devices. `tag/list` retrieves a list of user-defined tags for devices. `policy/recommendation` retrieves all active policy rule recommendations or those for one or more device profiles.

Parameter	Example
	When retrieving a list of items or details for a single item, the properties can be in any order within the returned JSON object.
General parameters	**`customerid=acmecorp`** indicates the customer ID. **`offset=1`** is an optional integer that sets the number of items to skip. **`pagelength=20`** is an optional integer that sets the number of items in one response; that is, in one page. The maximum page length you can set is 1000. The default page length for alerts, devices, and vulnerability instances is 1000. Because of these high default values, we recommend setting the page length to a smaller number, especially for alerts and vulnerabilities.
Device-specific parameters	**`deviceid=34:02:86:44:65:36`** specifies the MAC address of a device. For a static IP device, the device ID is its static IP address. **`ip=192.168.10.121`** specifies the IP address of a device. **`detail=false`** is an optional Boolean value requesting the level of device details to be returned. The default is **`false`**. **`detail=true`** enters detail mode, which returns more device properties; for example: **`‘https://acmecorp.iot.paloaltonetworks.com/pub/v4.0/device?detail=true&customerid=acmecorp’`** **`stime=2020-11-3T08:00Z`** is an optional string that sets the start of a time range for devices to retrieve. This is the time when a device was last active. (It’s unnecessary to set **`etime=now`** or **`etime=<time>`** because it is always treated as **`now`**.) **`sortdirection=asc`** is an optional string that sets the alphanumeric order in which devices are displayed by MAC address. **`asc`** indicates an ascending order from smallest to

Parameter	Example
	largest. **desc**, which is the default, indicates a descending order from largest to smallest. **sortfield=MAC** is an optional string that sets the field by which returned devices are sorted. Currently only **MAC** is supported. *You can* Use Queries from the IoT Security Portal *to customize which devices are retrieved.*
Alert-specific parameters	**type=policy_alert** is an optional string that returns security alerts. This is the only type of alert supported. **resolved=yes** is an optional string that returns only resolved alerts. **no** is the default and returns only active alerts. **stime=2020-11-3T08:00Z** is an optional string that sets the start of a time range for alerts to retrieve. (It's unnecessary to set **etime=now** or **etime=**<time> because it is always treated as now.) **sortdirection=asc** is an optional string that sets the chronological order in which alerts are displayed. **asc** is from oldest to newest. **desc** is from newest to oldest and is the default. **sortfield=date** is an optional string that sets the field by which returned alerts are sorted. Currently only **date** is supported. *You can* Use Queries from the IoT Security Portal *to customize which security alerts are retrieved.*
Vulnerability-specific parameters	**name=CVE-2018-18568** is an optional string that retrieves all instances of a specific vulnerability among your devices. **status=Confirmed** is an optional string that retrieves only confirmed vulnerabilities. **Potential** retrieves potential but unconfirmed vulnerabilities. If no value is passed, both types of vulnerabilities are retrieved.

Parameter	Example
	`groupby` is a required string. It specifies how to group device vulnerability instances in query results: **`groupby=device`** groups results by device ID. Each device ID and a single vulnerability are an item in the items list. **`groupby=vulnerability`** (the default) groups results by vulnerability. Each vulnerability and the device IDs impacted are an item in the items list. *You can Use Queries from the IoT Security Portal to customize which vulnerability instances are retrieved.*

Authentication and authorization	IoT Security issues the API Access Key and its ID. To authenticate and authorize your requests, pass the access key and its ID by adding two extra request headers:
	`X-Key-ID: KEY_ID`
	`X-Access-Key: ACCESS_KEY`
	For your requests to be authorized, the access key must be active and the user who created the key must be an owner or administrator.

*To prevent DoS (denial-of-service) attacks on our system, IoT Security imposes rate limits. When queries are for **`device/list`**, the rate limit is a maximum of 60 queries per minute per tenant because of the intensive amount of data that can potentially be returned. For everything else, the rate limit is 180 queries per minute.*

Before you can begin using the IoT Security API, you must generate the following from the IoT Security app:

- API Access Key
- API Key ID

Value	Description
API Access Key	The API Access Key is your unique token that's used as the **`"X-Access-Key: ACCESS_KEY"`** request header required for authenticating API calls.

Value	Description
API Key ID	The API Key ID is your unique identifier used to authenticate the API Access Key. The request header that's used when running an API call is **`"X-Key-Id: KEY_ID"`**.

The following steps describe how to generate the necessary key values.

STEP 1 | Log in to the IoT Security portal and click 👤 > **Preferences**.

STEP 2 | In the User Role & Access section, click **Create** next to API Access Key and follow the online steps to create an access key.

STEP 3 | View and download the access key and key ID, saving them in a secure location. Your code must include both when making calls to the API.

You can later return to the Preferences page to view the key ID. However, for security reasons, it is not possible to view the actual key in the IoT Security portal.

Use Queries from the IoT Security Portal

You can copy a query from the IoT Security portal, convert it to an ASCII string, and paste it in API requests to get customized lists of devices, vulnerability instances, and security alerts.

1. Log in to the IoT Security portal and open one of the following pages:

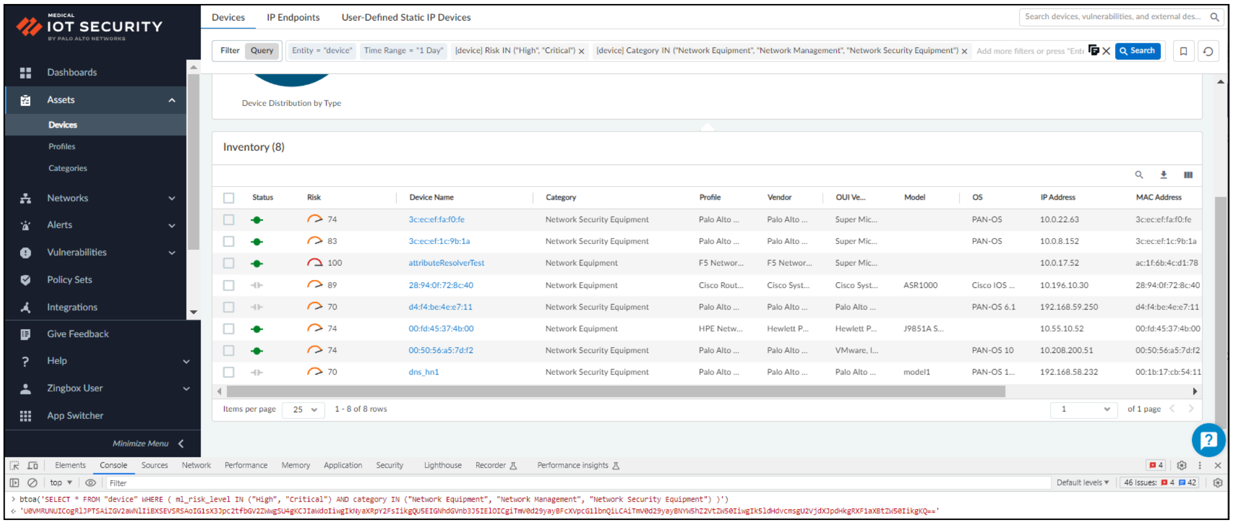
   Assets > **Devices**

   Vulnerabilities > **Vulnerability Overview**

   Alerts > **Security Alerts**

2. At the top of the page, click **Query** and use the query builder to define a query to fetch the list of items you want to see.

   For example, the following query on the **Assets** > **Devices** page gets a list of various network devices at high or critical risk levels.



Entity = "device" | Time Range = "1 Day" | [device] Risk IN ("High", "Critical") × | [device] Category IN ("Network Equipment", "Network Management", "Network Security Equipment") × | Add more filters or press "Ente

Inventory (8)

□	Status	Risk	Device Name	Category	Profile	Vendor	OUI Ve...	Model	OS	IP Address	MAC Address
□	●	74	3c:ec:ef:fa:f0:fe	Network Security Equipment	Palo Alto ...	Palo Alto ...	Super Mic...		PAN-OS	10.0.22.63	3c:ec:ef:fa:f0:fe
□	●	83	3c:ec:ef:1c:9b:1a	Network Security Equipment	Palo Alto ...	Palo Alto ...	Super Mic...		PAN-OS	10.0.8.152	3c:ec:ef:1c:9b:1a
□	●	100	attributeResolverTest	Network Equipment	F5 Networ...	F5 Networ...	Super Mic...			10.0.17.52	ac:1f:6b:4c:d1:78
□	○	89	28:94:0f:72:8c:40	Network Equipment	Cisco Rout...	Cisco Syst...	Cisco Syst...	ASR1000	Cisco IOS ...	10.196.10.30	28:94:0f:72:8c:40
□	○	70	d4:f4:be:4e:e7:11	Network Security Equipment	Palo Alto ...	Palo Alto ...	Palo Alto ...		PAN-OS 6.1	192.168.59.250	d4:f4:be:4e:e7:11
□	●	74	00:fd:45:37:4b:00	Network Equipment	HPE Netw...	Hewlett P...	Hewlett P...	J9851A S...		10.55.10.52	00:fd:45:37:4b:00
□	●	74	00:50:56:a5:7d:f2	Network Security Equipment	Palo Alto ...	Palo Alto ...	VMware, I...		PAN-OS 10	10.208.200.51	00:50:56:a5:7d:f2
□	○	70	dns_hn1	Network Security Equipment	Palo Alto ...	Palo Alto ...	Palo Alto ...	model1	PAN-OS 1...	192.168.58.232	00:1b:17:cb:54:11

Items per page 25 1 - 8 of 8 rows 1 of 1 page

```
> btoa('SELECT * FROM "device" WHERE ( ml_risk_level IN ("High", "Critical") AND category IN ("Network Equipment", "Network Management", "Network Security Equipment") )')
<· 'U0VMRUNUICogRlJPTSAiZGV2aWNlIiBXSEVSRSAoIG1sX3Jpc2tfbGV2ZWwgSU4gKCJIaWdoIiwgIkNyaXRpY2FsIikgQU5EIGNhdGVnb3J5IElOICgiTmV0d29yayBFcXVpcGllbnQiLCAiTmV0d29yayBNYW5hZ2VtZW50IiwgIk5ldHdvcmsgU2VjdXJpdHkgRXF1aXBtZW50IikgKQ=='
```

3. Click the **Copy to clipboard** icon (⧉) at the far right of the Query builder field.

4. Open a JavaScript console and use the `btoa` command to convert the binary string to a Base64-encoded ASCII string.

   For example, in a Chrome browser, click the three vertical dots icon **(⋮)** > **More tools** > **Developer tools**, click the **Console** tab, and then enter:

   ```
   btoa('SELECT * FROM "device" WHERE ( ml_risk_level IN ("High",
    "Critical") AND
           category IN ("Network Equipment", "Network Management",
    "Network Security Equipment")
           )')
   ```

   This produces the following ASCII string:

   ```
   'U0VMRUNUICogRlJPTSAiZGV2aWNlIiBXSEVSRSAoIG1sX3Jpc2tfbGV2ZWwgSU4gKCJIaWdoIiwg
   ```

5. Copy the ASCII string, excluding the single quotation marks and paste it into the `get` request as a SQL string (`sql_str` parameter) after the tenant name.

   When you copy a query from the IoT Security portal, the time filters that you see on the page aren't included. By default, an API request using the query will get all results to date. If you want to include a time filter, use the `stime` parameter as shown in the example below.

   Example:

   ```
   curl --location -X GET 'http://acmecorp.iot.paloaltonetworks.com/
   pub/v4.0/device/list?customerid=staging-banff-
   test&stime=2023-07-01T08:00Z&sql_str=U0VMRUNUICogRlJPTSAiZGV2aWNlIiBXSEVSRSAo
   -H 'X-Key-Id: KEY_ID' \
   -H 'X-Access-Key: ACCESS_KEY'
   ```

   When requesting a device list, use only `[device]` options in the query builder; do not include any `[vulnerability]` or `[alert]` options. Although the IoT Security portal UI supports queries that combine devices and security alerts or devices and vulnerabilities, the API doesn't support these combinations.

IoT Security API

Get Device Details per Device ID

Synopsis

URI	/pub/v4.0/device
HTTP Method	GET
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Get a list of device details for the device with the specified device ID. The device ID is typically a MAC address, but an IP address is used for devices configured as static IP devices.

Request Fields

The URL of this request contains the following parameters:

Field	Description
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.
`deviceid`	(Required) The device ID specifies the MAC address of the device for which you want to get details. It's an IP address when the device is configured as a static IP device. The following value is a string.

For additional common parameters you can use with this request, check Get Started with the IoT Security API.

Request Example

```
curl --location -X GET 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/device?customerid=acmecorp&deviceid=34:02:86:44:65:36' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

The `--location` option is necessary because some API requests elicit a 3xx response code, indicating that redirection to another destination is required to fetch the requested data, and the `--location` option enables curl to redo the request to the new destination.

Success Response

Upon success, the HTTP response code is 200. In addition, this API returns a JSON object containing an array of JSON objects, each of which represents a single device attribute.

Field	Description
`deviceid`	The device ID, which IoT Security uses to identify and track the device (string)
`hostname`	The device hostname (string)
`category`	The category to which the device belongs (string)
`profile`	The device profile assigned to the device (string)
`profile_type`	The type of device profile, such as **`IoT`** or **`Non_IoT`** (string)
`profile_vertical`	The industry vertical for the profile such as **`Medical`**, **`IT Devices`**, and **`Office`** (string)
`ip_address`	The IP address of the device (string)
`mac_address`	The MAC address of the device (string)
`tagIdList`	A list of IDs for user- and system-defined tags assigned to the device
`risk_score`	The risk score of the device (integer)
`risk_level`	The risk level of the device; there are four: **`low`**, **`medium`**, **`high`**, and **`critical`** (string)
`last_activity`	A UTC timestamp for the last detected device activity (object)
`confidence_score`	The confidence score for device classification (integer)
`subnet`	The subnet to which the device is attached (string)
`number_of_critical_alerts`	The number of critical alerts for the device (integer)
`number_of_warning_alerts`	The number of warning alerts for the device (integer)

Field	Description
`number_of_caution_alerts`	The number of caution alerts for the device (integer)
`number_of_info_alerts`	The number of info alerts for the device (integer)
`allTags`	An array of user-defined tags assigned to the device. Each item in the array consists of three attributes: `tagType`, `tagValue`, and `tagId`.
`tagType`	The key for a user-defined tag
`tagValue`	The value of the tag key for a user-defined tag
`tagId`	The ID of a user-defined tag

Success Response Example

```
{
    "deviceid": "34:02:86:44:65:36",
    "hostname": "InfusionPump-20",
    "category": "Infusion System",
    "profile": "Sigma Spectrum Infusion System",
    "profile_type": "IoT",
    "profile_vertical": "Medical",
    "ip_address": "192.168.10.121",
    "mac_address": "34:02:86:44:65:36",
    "tagIdList":
    [
      "6030135777a1d6fb488e26ad",
      "60301332ff1679e9481b62a6",
      "602ca12179bc780a2333895d",
    ],
    "risk_score": 0,
    "risk_level": "low",
    "last_activity": "2018-05-31T18:39:37.404Z",
    "confidence_score": 90,
    "subnet": "192.168.10.121/28",
    "number_of_critical_alerts": 0,
    "number_of_warning_alerts": 0,
    "number_of_caution_alerts": 0,
    "number_of_info_alerts": 0,
    "allTags":
    [
      {
      "tagType": "infusion",
      "tagValue": "pump1",
      "tagId": "6030135777a1d6fb488e26ad",
      },
      {
      "tagType": "infusion",
      "tagValue": "pump2",
      "tagId": "60301332ff1679e9481b62a6",
      },
      {
      "tagType": "infusion",
      "tagValue": "pump3",
      "tagId": "60f221a219e22f10003a965e",
      },
    ],
    ...
}
```

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains an invalid query string.
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests for device details for a single device exceeded the rate limit of 180 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```

Get Device Details per IP Address

Synopsis

URI	/pub/v4.0/device/ip
HTTP Method	GET
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Get a list of device details for the device with the specified IP address.

Request Fields

The URL of this request contains the following parameters:

Field	Description
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.
`ip`	(Required) This is the IP address of the device for which you want to get details. The following value is a string.

For additional common parameters you can use with this request, check Get Started with the IoT Security API.

Request Example

```
curl --location -X GET 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/device/ip?customerid=acmecorp&ip=192.168.10.121' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

Success Response

Upon success, the HTTP response code is 200. In addition, this API returns a JSON object containing an array of JSON objects, each of which represents a single device attribute.

Field	Description
`deviceid`	The device ID, which IoT Security uses to identify and track the device (string)
`hostname`	The device hostname (string)
`category`	The category to which the device belongs (string)
`profile`	The device profile assigned to the device (string)
`profile_type`	The type of device profile, such as **`IoT`** or **`Non_IoT`** (string)
`profile_vertical`	The industry vertical for the profile such as **`Medical`**, **`IT Devices`**, and **`Office`** (string)
`ip_address`	The IP address of the device (string)
`mac_address`	The MAC address of the device (string)
`risk_score`	The risk score of the device (integer)
`risk_level`	The risk level of the device; there are four: **`low`**, **`medium`**, **`high`**, and **`critical`** (string)
`last_activity`	A UTC timestamp for the last detected device activity (object)
`confidence_score`	The confidence score for device classification (integer)
`subnet`	The subnet to which the device is attached (string)
`number_of_critical_alerts`	The number of critical alerts for the device (integer)
`number_of_warning_alerts`	The number of warning alerts for the device (integer)
`number_of_caution_alerts`	The number of caution alerts for the device (integer)
`number_of_info_alerts`	The number of info alerts for the device (integer)
`tagIdList`	A list of IDs for user- and system-defined tags assigned to the device

Success Response Example

```
{
    "deviceid": "34:02:86:44:65:36",
    "hostname": "InfusionPump-20",
    "category": "Infusion System",
    "profile": "Sigma Spectrum Infusion System",
    "profile_type": "IoT",
    "profile_vertical": "Medical",
    "ip_address": "192.168.10.121",
    "mac_address": "34:02:86:44:65:36",
    "risk_score": 0,
    "risk_level": "low",
    "last_activity": "2018-05-31T18:39:37.404Z",
    "confidence_score": 90,
    "subnet": "192.168.10.121/28",
    "number_of_critical_alerts": 0,
    "number_of_warning_alerts": 0,
    "number_of_caution_alerts": 0,
    "number_of_info_alerts": 0,
    "tagIdList":
    ...
    }
```

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains an invalid query string.
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests for device details for a single device exceeded the rate limit of 180 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```

Get the Device Inventory

Synopsis

URI	/pub/v4.0/device/list
HTTP Method	GET
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Get a list of all the devices in your IoT Security inventory.

Request Fields

The URL of this request contains the following parameters:

Field	Description
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.

For other parameters you can include in the URL—such as `offset`, `pagelength`, `sortdirection`, `sortfield` and `stime`—see the general parameters and device-specific parameters described in Get Started with the IoT Security API. *You can also* Use Queries from the IoT Security Portal *to customize which devices are retrieved.*

Request Example

```
curl --location -X GET 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/device/list?customerid=acmecorp' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

Success Response

Upon success, the HTTP response code is 200. In addition, this API returns a JSON object containing devices and their attributes.

Field	Description
`total`	The number of devices matching the request
`devices`	An array containing device details

Field	Description
`deviceid`	The device ID, which IoT Security uses to identify and track a device (string)
`hostname`	Device hostname (string)
`category`	Category to which a device belongs (string)
`profile`	Device profile assigned to a device (string)
`profile_type`	Type of device profile, such as **`IoT`** or **`Non_IoT`** (string)
`profile_vertical`	Industry vertical for a device profile such as **`Medical`**, **`IT Devices`**, and **`Office`** (string)
`ip_address`	IP address of a device (string)
`mac_address`	MAC address of a device (string)
`risk_score`	Risk score of a device (integer)
`risk_level`	Risk level of a device; there are four: **`low`**, **`medium`**, **`high`**, and **`critical`** (string)
`last_activity`	UTC timestamp for the last detected device activity (object)
`confidence_score`	Confidence score for device classification (integer)
`trafficRestricted`	Whether traffic restriction is being applied to a device (**`yes`**) or not (**`no`**)
`tagIdList`	A list of IDs for user- and system-defined tags assigned to a device
`allTags`	An array of user-defined tags assigned to a device. Each item in the array consists of three attributes: `tagType`, `tagValue`, and `tagId`.
`tagType`	The key for a user-defined tag
`tagValue`	The value of the key for a user-defined tag
`tagId`	The ID of a user-defined tag
`total`	The total number of devices for which information was returned

To get more attributes for each device, include **`detail=true`** *in the request. See Device-specific parameters in* Get Started with the IoT Security API.

Success Response Example

```
{
    "devices": [
      {
      "deviceid": "34:02:86:44:65:36",
      "hostname": "InfusionPump-20",
      "last_activity": "2018-05-31T18:39:37.404Z",
      "category": "Infusion System",
      "profile": "Sigma Spectrum Infusion System",
      "profile_type": "IoT",
      "profile_vertical": "Medical",
      "ip_address": "192.168.10.121",
      "mac_address": "34:02:86:44:65:36",
      "risk_score": 0,
      "risk_level": "low",
      "confidence_score": 90},
      "trafficRestricted": "no",
      "tagIdList": [
        "60f221a219e22f10003a965e"
        ],
      "allTags":
        [
          {
          "tagType": "med-equipment",
          "tagValue": "infusion",
          "tagId": "60f221a219e22f10003a965e"
          }
        ]
      }
      ...
    ],
    "total": 100
}
```

Data is shown for only the first of 100 devices in the full response, and detail mode is off.

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains an invalid query string.

Field	Description
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests for the device inventory list exceeded the rate limit of 60 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```

Get Profile Mapping

Synopsis

URI	/pub/v4.0/profile/mapping
HTTP Method	GET
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Get a list of device profiles with each profile mapped to a category and vertical.

This is not a list of mappings for just the device profiles in your environment but for all device profiles that can appear in IoT Security.

Request Fields

The URL of this request contains the following parameter:

Field	Description
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.

Request to Get a List of Profile Mappings Example

```
curl 'https://acmecorp.iot.paloaltonetworks.com/pub/v4.0/profile/
mapping?customerid=acmecorp' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

Success Response

Upon success, the HTTP response code is 200.

Success Response Example

```
{
    "mapping": [
        {
            "profile": "Formlabs 3D Printer",
            "category": "3D Printer",
            "vertical": "Office"
        },
        {
            "profile": "Edgewater Networks Device",
            "category": "Network Equipment",
            "vertical": "Network Devices"
        },
        {
            "profile": "Naim Speaker",
            "category": "Smart Speaker",
            "vertical": "Consumer IoT"
        }
        …
    ],
    "count": 3041
}
```

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains an invalid query string.
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests for a list of profile mappings exceeded the rate limit of 180 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```

Get Security Alerts

Synopsis

URI	/pub/v4.0/alert/list
HTTP Method	GET
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Get a list of security alerts.

Request Fields

The URL of this request contains the following parameters:

Field	Description
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.
`type`	Optional field specifying the alert type as **`policy_alert`**. The following value is a string.
`resolved`	Optional field to get only active alerts (**`resolved=no`**) or resolved alerts (**`resolved=yes`**). The default is to get both types of alerts. The following value is a string.
`pagelength`	Optional but recommended field specifying the number of items for each page. The default page length for alerts is 1000 and the maximum is 1000. Setting a shorter length improves response times. The following value is an integer.
`offset`	In addition to the **`pagelength`** parameter, use **`offset`** to get items on subsequent pages. For example, if your first request is **`pagelength = 100`**, you will get the first 100 device alerts. To get the next 100, add **`offset = 100`** to

Field	Description
	your second request. This skips the first 100 alerts and gets the next 100 starting from 101.
`stime`	Optional string that sets the start of a time range for alerts to retrieve. For example, **`stime=2021-10-6T07:00Z`**. (It's unnecessary to set **`etime=now`** or **`etime=`**`<time>` because it is always treated as `now`.)
`sortdirection`	Optional field defining whether the alerts are organized in ascending order **`asc`** (oldest to newest) or descending order **`desc`** (newest to oldest). The default is **`desc`**. The following value is a string.
`sortfield`	Optional field that defines how alerts are ordered. **`date`** and **`severityNumber`** are supported as the following value and the value types are **`string`** and **`integer`** respectively. The default way to sort alerts is by date in descending order.

You can also Use Queries from the IoT Security Portal *to customize which security alerts are retrieved.*

Request Example

```
curl --location -X GET  'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/alert/list?
customerid=acmecorp&type=policy_alert&resolved=no&pagelength=1&sortdirection=de
 \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

Success Response

Upon success, the HTTP response code is 200. In addition, this API returns a JSON object containing an array of JSON objects, representing devices and their attributes.

Field	Description
`resolved`	Whether the alert has been resolved **`yes`**or not **`no`** (string)
`date`	The alert detection date

Field	Description
`deviceid`	The MAC address or IP address of a device (string)
`name`	The alert name (string)
`severity`	The severity level of an alert: high, medium, low, info (string)
`severityNumber`	The severity number matching the severity level: high = 4, medium = 3, low = 2, info = 1 (integer)
`type`	The type of alert (string)
`description`	The alert description (string)
`tenantid`	The internal customer ID (string)
`zb_ticketid`	The unique ticket ID (integer)
`id`	The alert ID. This is the ID to use when resolving an alert through the API (integer)
`profile`	The device profile to which the alert belongs (string)
`profile_vertical`	The industry vertical for the profile such as Medical, IT Devices, and Office.
`category`	The device category to which the alert belongs (string)
`hostname`	The hostname of the device to which the alert belongs (string)
`siteid`	The ID number that IoT Security assigns to the site for internal use (string)
`serviceLevel`	(For MSSP only) The level of service for an MSSP customer as defined by the MSSP owner; for example: `Tier 1, Tier 2, Tier 3`; or `Platinum, Gold, Silver` (string)
`trafficDirection`	The direction of the traffic on the device that triggered the alert; `inbound` if the device is a server and `outbound` if it is a client (string)

Field	Description
`siteName`	The name of the site where the alert occurred (string)
`reason_history`	The history of actions taken to investigate and resolve the alert (string)
`total`	The overall number of security alerts for all the IoT devices in your inventory

Success Response Example

```
{"ver": "v4.0",
    "api": "/alert/list",
    "items": [
         {
            "resolved": "no",
            "date": "2020-05-12T01:23:10.630Z",
            "deviceid": "18:65:90:cd:88:0d",
            "name": "zingcloud alert bg job integration test at
 1589246590630",
            "severity": "high",
            "severityNumber": 4,
            "type": "policy_alert",
            "description": "The baseline policy defines a criteria to
 match normal connections between devices in two different networks
 or device groups. It is a positive detection if connections outside
 of this definition are observed.",
            "tenantid": "acmecorp"
            "zb_ticketid": "alert-hNMleG1nW",
            "id": "5eb9fa8127b736d82bf7840a",
            "profile": "Macintosh-MacPro",
            "profile_vertical": "IT Devices",
            "category": "Personal Computer",
            "hostname": "cntl-201-2",
            "siteid": "0",
            "serviceLevel": "",
            "trafficDirection": "inbound",
            "siteName": "acmecorp-hq",
            "reason_history": []
            "msg": {
                "severity": "high",
                "description": "The baseline policy defines criteria
 to match normal connections between devices in two different
 networks or device groups. It is a positive detection if connections
 outside of this definition are observed.",
                "name": "zingcloud alert bg job integration test at
 1589246590630",
                "id": "hNMleG1nW",
                "ruleid": "5a26f169c8272f0b00c5ef1a",
                "zb_ticketid": "alert-hNMleG1nW",
                "hostname": "unknown",
                ...            }        },
                    . . .
    ],
    "total": 39
}
```

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains an invalid query string.
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests for a list of security alerts exceeded the rate limit of 180 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```

Resolve a Security Alert

Synopsis

URI	/pub/v4.0/alert/update
HTTP Method	PUT
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Resolve a security alert.

Request Fields

The URL of this request contains the following parameters:

Field	Description
`id`	(Required) The alert ID being resolved. To retrieve a list of security alerts, including their IDs, use Get Security Alerts. *Use the value for `id`, not the value for `zb_ticketid`.* The following value is a string.
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.

The payload of this request contains the following parameters:

Field	Description
`reason`	(Required) This is the reason for resolving the alert. The following value is a string and cannot contain any special characters.
`reason_type`	(Required) This is the type of reason for resolving the alert and is one of the following array of values: **`Issue Mitigated`**

Field	Description

	`No Action Needed`
	`VPN protected connections`
	`Trusted remote destination`
	`Normal behavior for this device`
	`Normal behavior for all devices in the same IoT profile`
	`Other`

`resolved`	(Required) This defines the alert as resolved. The following value is a string and must be **`yes`**.

Request to Resolve an Alert Example

```
curl --location -X PUT 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/alert/update?id=<alert_id_number>&customerid=acmecorp' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY' \
--header 'Content-Type: application/json' \
--data-raw \
'{
    "reason": "The alert poses no threat",
    "reason_type":
      [
      "No Action Needed"
      ],
    "resolved": "yes"
}'
```

Success Response

Upon success, the HTTP response code is 200.

Success Response Example

```
{
    "api": "/pub/v4.0/alert/update",
    "ver":"v0.3"
}
```

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains invalid JSON in its body.
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests to resolve a security alert exceeded the rate limit of 180 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```

Get Vulnerability Instances

Synopsis

URI	/pub/v4.0/vulnerability/list
HTTP Method	GET
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Get a list of device vulnerability instances.

Request Fields

The URL of this request contains the following parameters:

Field	Description
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.
`stime`	Optional field setting the start of a time range for retrieving vulnerability instances. For example, to get all instances since November 3, 2020 starting at 00:00 AM in the Pacific Time Zone (UTC-8), the start time would be **`stime=2020-11-3T08:00Z`**. If you prefer to specify the time in your local time rather than adjusting it to UTC time, you can also format the start time as **`2020-11-03T:00:00-08:00`**. Especially when starting at a later hour in the day, this format involves less calculating. For example, if you want to get vulnerability instances starting from 6:00 PM on November 3, 2020, entering **`2020-11-03T18:00-08:00`** is much simpler than entering **`2020-11-04T02:00Z`**.
`pagelength`	Optional but recommended field specifying the number of items for each page. The default page length for vulnerabilities is 1000 and the maximum is 1000. Setting a

Field	Description
	shorter length improves response times. The following value is an integer. 📋 *The `pagelength` parameter is only valid when grouping vulnerability instances by device, not when grouping them by vulnerability.*
`offset`	In addition to the **`pagelength`** parameter, use **`offset`** to get items on subsequent pages. For example, if your first request is **`pagelength = 100`**, you will get the first 100 vulnerabilities (indexed from 0 to 99). To get the next 100, add **`offset = 100`** to your second request. This skips the first 100 vulnerabilities and gets the next 100 starting from index number 100. 📋 *The `offset` parameter is only valid when grouping vulnerability instances by device, not when grouping them by vulnerability.*
`name`	Optional field defining a specific vulnerability. If omitted, instances for all vulnerabilities are returned. The following value is a string.
`status`	Optional field that retrieves only confirmed or potential vulnerability instances. The following field is either the string **`Confirmed`** or **`Potential`**. If no value is passed, both types of vulnerabilities are returned.
`groupby`	(Required) This specifies how to group device vulnerability instances in query results. Each `groupby` option results in a different JSON object structure in the response. **`groupby=vulnerability`** (the default) organizes results into groups by vulnerability. Each vulnerability and the device IDs impacted are an item in the items list. **`groupby=device`** organizes results into groups by device ID. Each device ID and a single vulnerability are an item in the items list.

Field	Description
	To request all vulnerability instances for a specific device, the value is the string **`vulnerability`** followed by **`&deviceid=<device_id>`**, where the device ID is either a MAC address or, for static IP devices, an IP address. (Entering an IP address for a device whose device identifier is a MAC address doesn't work. Similarly, entering a MAC address for a device whose device identifier is an IP address also doesn't work.)

You can also Use Queries from the IoT Security Portal to customize which vulnerability instances are retrieved.

Request All Vulnerability Instances Grouped by Device Example

```
curl --location -X GET 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/vulnerability/list?customerid=acmecorp&groupby=device' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

Request All Vulnerability Instances for a Specific Device Example

```
curl --location -X GET 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/vulnerability/list?
customerid=acmecorp&groupby=device&deviceid=64:16:7f:0a:f6:38' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

Request All Vulnerability Instances Grouped by Vulnerability Example

```
curl --location -X GET 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/vulnerability/list?
customerid=acmecorp&groupby=vulnerability' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

Success Response

Upon success, the HTTP response code is 200. In addition, this API returns a JSON object containing an array of JSON objects, representing devices and their attributes.

The fields returned differ depending on whether you group results by device or by vulnerability. Both sets of fields are shown below.

When the request includes **`groupby=device`**, the response includes the following fields:

Field	Description
`items`	Introduces items in the list of vulnerability instances
`name`	The hostname of the device associated with a vulnerability instance (string)
`ip`	The IP address of the device associated with a vulnerability instance (string)
`deviceid`	The MAC address or IP address of the device (string)
`profile`	The profile to which the device belongs (string)
`profile_vertical`	The vertical to which the device profile belongs (string)
`display_profile_category`	The category to which the device profile belongs (string)
`vendor`	The device vendor (string)
`model`	The device model (string)
`os`	The device OS (sting)
`osCombined`	The OS and OS version combined (string)
`siteid`	The ID of the site where the device is deployed (string)
`asset_tag`	The asset tag of the device; if none, then “null” is returned (string)
`sn`	The device serial number (string)
`date`	The date of the latest activity of the device (string)
`risk_score`	The risk score of the vulnerability instance (integer)
`risk_level`	The risk level of the vulnerability instance: Low, Medium, High, or Critical (string)
`ticketState`	The state of the zb_ticket for a vulnerability instance—**`investigation`**, **`remediation`**,

Field	Description
	resolved, or **new** if the vulnerability was detected but nobody has yet taken action to address it (string)
zb_ticketid	The unique ticket ID for a vulnerability instance (integer)
ticketAssignees	The email address of one or more people assigned to remediate a vulnerability instance; if there aren't any, **null** is returned (string)
reason_history	An array that holds the history of all actions taken on a vulnerability instance, including status changes, user notes, if it was sent to asset management, and if it was resolved; if no actions were taken, **null** is returned (string)
remediate_workorder	The work order number returned from an integrated third-party asset management system such as AIMS, Connectiv, Nuvolo, or ServiceNow to which a vulnerability instance was sent (string)
remediate_checkbox	Index values indicating the type of information sent to asset management; 0 = vulnerability summary, 1 = vulnerability impact, 2 = device information
remediate_instruction	Additional instructions included with the work order (string)
detected_date	The date when a vulnerability instance was originally detected (string)
vulnerability_name	The name of the vulnerability (string)
tagIdList	A list of IDs for user- and system-defined tags assigned to a device
allTags	An array of user-defined tags assigned to the device. Each item in the array consists of three attributes: tagType, tagValue, and tagId.
tagType	The key for a user-defined tag
tagValue	The value of the key for a user-defined tag

Field	Description
`tagId`	The ID of a user-defined tag
`total`	The total overall number of vulnerability instances on the devices in your network

When the request includes **`groupby=vulnerability`**, the response includes the following fields:

Field	Description
`items`	Introduces items in the list of vulnerability instances
`data`	The device profiles, IoT devices, and number of addressed instances for confirmed and potential instances in a vulnerability group
`Potential`	The following data is for potential vulnerability instances
`Confirmed`	The following data is for confirmed vulnerability instances
`profile`	The device profiles that the vulnerability affects or potentially affects
`device`	List of device IDs of all IoT devices that are vulnerable or potentially vulnerable
`addressedInstance`	The number of instances that have been addressed
`name`	Name of the vulnerability; for example, CVE-2015-3959 or Windows SMBv1 Usage
`cvssVersion`	**`v2`** or **`v3`** Score ranges are different for the two Common Vulnerability Scoring System versions. Version 2 has three score ranges: Low 0.0 - 3.9, Medium 4.0 - 6.9, High 7.0 - 10.0. Version 3 has five ranges: None - 0.0, Low 0.1 - 3.9, Medium 4.0 - 6.9, High, 7.0 - 8.9, Critical 9.0 - 10.0.)
`severity`	**`Low`**, **`Medium`**, **`High`**, **`Critical`** Only vulnerabilities with a CVSSv3 rating can have a Critical severity level.

Field	Description
`date`	The most recent date that an instance of this vulnerability was detected on an IoT device
`CVSS`	The CVSS score for the vulnerability; for example, 10.0
`description`	A description of what the vulnerability is and how it can be exploited
`deviceid`	The MAC address or IP address of a device that either has or potentially has a vulnerability
`source`	The source of the vulnerability detection: `IoT Security` when IoT Security learns it through its own detection and analysis; or, if learned through integration with a third-party vulnerability scanner, the name of the scanner-`Qualys`, `Rapid7`, `Tenable`
`vulnerability_types`	The type of attack to which a vulnerability makes a device susceptible; for example, `Code Execution`, `Overflow`, `Info Leak`, `Denial of Service`
`deviceTags`	A list of device IDs of all vulnerable and potentially vulnerable IoT devices and any tags they might have
`allTags`	An array of user-defined tags assigned to the device. Each item in the array consists of three attributes: `tagType`, `tagValue`, and `tagId`.
`tagType`	The key for a user-defined tag
`tagValue`	The value of the key for a user-defined tag
`tagId`	The ID of a user-defined tag
`source`	The source of the vulnerability detection: `IoT Security` when IoT Security learns it through its own detection and analysis; or, if learned through integration with a third-party vulnerability scanner, the name of the scanner-`Qualys`, `Rapid7`, `Tenable`

Field	Description
`total`	The total number of vulnerabilities affecting devices in your network

Success Response for All Vulnerability Instances (**`groupby=device`**) Example

```
{
"ver": "v4.0",
"api": "/vulnerability/list",
"items": [
      {
      "deviceid": "64:16:7f:37:2d:45",
      "detected_date": [
        "2021-04-19T23:59:59"
        ],
      "name": "Polycom_64167f372d45",
      "ip": "10.1.1.84",
      "profile": "Polycom IP Phone",
      "profile_vertical": "Office",
      "display_profile_category": "IP Phone",
      "vendor": "Polycom",
      "model": "VVX601",
      "os": "Embedded",
      "osCombined": "Embedded",
      "siteid": "0",
      "asset_tag": null,
      "sn": null,
      "date": "2021-03-12T01:28:26.986Z",
      "risk_score": 20,
      "risk_level": "Low",
      "tagIdList": [
        "6030135777a1d6fb488e26ad",
        "60301332ff1679e9481b62a6"
      ],
      "ticketState": "new",
      "zb_ticketid": "vuln-52f40a58",
      "ticketAssignees": [
        "analyst1@acmecorp.com"
      ],
      "reason_history": [
        {
        "action": "sent to asset management: aims",
        "reason": "Check system",
        "reason_type": null,
        "user_email": "admin@acmecorp.com",
        "timestamp": "2019-10-18T22:00:20.255Z",
        "aims_workorder_number": 152027
        "remediate_workorder": "152027",
        "remediate_checkbox": "0,1,2",
        "remediate_instruction": null,
        "detected_date": "2019-10-15T20:18:42.135Z",
        "vulnerability_name": "CVE-2019-12948"
        },
      ],
      "allTags": [
        {
        "tagType": "Owner",
        "tagValue": "Joe",
        "tagId": "6030135777a1d6fb488e26ad"
        },
        {
        "tagType": "com-devices",
        "tagValue": "phones",
        "tagId": "60301332ff1679e9481b62a6"
        },
      ]
   }
   ...
   ]
   "total": 34,
}
```

Success Response for All Vulnerability Instances (**`groupby=vulnerability`**) Example

```
{
    "ver": "v4.0",
    "api": "/vulnerability/list",
    "items": {
        "items": [
            {
                "data": {
                    "Potential": {
                        "profile": [
                            "Arista Networks Device",
                            "Cisco Systems Device"
                        ],
                        "device": [
                            "00:1c:73:20:c4:b5",
                            "00:1c:73:16:a6:33",
                            "00:57:d2:27:d2:d1"
                        ],
                        "addressedInstance": 0
                    }
                    "Confirmed": {
                        "profile": [
                            "Roles-Network-Router",
                            "Cisco Networking Switch"
                        ],
                        "device": [
                            "e4:d3:f1:40:e8:c0",
                            "08:17:35:77:d0:c1"
                        ],
                        "addressedInstance": 0
                    }
                },
                "name": "CVE-2019-1737",
                "tenantid": "",
                "cvssVersion": "v3",
                "serviceLevel": null,
                "severity": "High",
                "date": "2022-07-18T23:59:59.000Z",
                "CVSS": 8.6,
                "description": "A vulnerability in the processing
 of IP Service Level Agreement (SLA) packets by Cisco IOS Software
 and Cisco IOS XE software could allow an unauthenticated, remote
 attacker to cause an interface wedge and an eventual denial of
 service (DoS) condition on the affected device. The vulnerability
 is due to improper socket resources handling in the IP SLA responder
 application code. An attacker could exploit this vulnerability by
 sending crafted IP SLA packets to an affected device. An exploit
 could allow the attacker to cause an interface to become wedged,
 resulting in an eventual denial of service (DoS) condition on the
 affected device.",
                "source": "IoT Security",
                "vulnerability_types": [
                    "Denial Of Service"
                ]
            },
            ...
      ]
      "deviceTags": {
            "08:17:35:77:d0:c1": {
                "deviceid": "08:17:35:77:d0:c1",
                "allTags": [
                    {
                        "tagType": "lab",
                        "tagValue": "l3",
                        "tagId": "62acbf20a5fb040006174076"
                    },
                    {
                        "tagType": "location",
                        "tagValue": "office14",
                        "tagId": "62acbf74a5fb040006174078"
                    },
                    {
                        "tagType": "Forescout",
                        "tagValue": "In Scope",
                        "tagId": "6144c5700a5895bbb5384b88"
                    },
                ]
            },
            "00:0f:11:00:e9:f2": {
                "deviceid": "00:0f:11:00:e9:f2"
            },
            "00:a0:aa:00:01:96": {
                "deviceid": "00:a0:aa:00:01:96"
            },
            ...
        }
    }
    "total": 61
}
```

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains an invalid query string.
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests for a list of vulnerability instances exceeded the rate limit of 180 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```

Resolve Vulnerability Instances

Synopsis

URI	/pub/v4.0/vulnerability/update
HTTP Method	PUT
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Mark one or more instances of a vulnerability as resolved.

Request Fields

The URL of this request contains the following parameter:

Field	Description
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.

The payload of this request contains the following parameters:

Field	Description
`ticketIdList`	(Required) This is a list of one or more ticket IDs for vulnerability instances being resolved. If you include the ticket IDs for multiple vulnerability instances, separate them by commas; for example: **`"ticketIdList": ["vuln-1a4a72c2", "vuln-1a4a72c3", "vuln-1a4a72c4"]`** To retrieve a list of vulnerabilities and vulnerability instances, including their ticket IDs, use Get Vulnerability Instances and refer to the **`zb_ticketid`** values.
`action`	(Required) This is the action employed to resolve the vulnerability instance. The following value is a string and must be either **`mitigate`** or **`ignore`**.

Field	Description
reason	(Required) This is the reason for resolving the vulnerability instance. The following value is a string and cannot contain any special characters.
full_name	(Required) This is the name of the vulnerability; for example, **CVE-2018-18568**. The following value is a string.

Request to Resolve a Vulnerability Instance Example

```
curl --location -X PUT 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/vulnerability/update?customerid=acmecorp' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY' \
--header 'Content-Type: application/json' \
--data-raw \
'{
    "ticketIdList":
      [
      "vuln-1a4a72c2"
      ],
    "action": "mitigate",
    "reason": "Threat was removed",
    "full_name": "CVE-2018-18568"
}'
```

Success Response

Upon success, the HTTP response code is 200.

Success Response Example

```
{
    "api": "/pub/v4.0/vulnerability/update",
    "ver":"v4.0",
    "updatedVulnerInstanceList":
      [
      "newScore": 18,
      "newLevel": "Low",
      "newAnomalyMap":
          {
          "application": 0,
          "payload": 0,
          "internal": 0,
          "external": 0,
          "protocol": 0
          }
      ]
}
```

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains invalid JSON in its body.
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests to resolve a vulnerability instance exceeded the rate limit of 180 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```

Add User-defined Tags

Synopsis

URI	/pub/v4.0/device/update
HTTP Method	PUT
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Add a user-defined tag to one or more IoT devices.

Request Fields

The URL of this request contains the following parameters:

Field	Description
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.

The payload of this request contains the following parameters:

Field	Description

`tag`	A user-defined tag being assigned to one or more devices. The tag can be a string or object, cannot contain special characters, and can contain one or two attributes: `tagType` (optional) and `tagValue`.
	If you enter both components of a key-value pair, then you must use the keyword for each field. Example:
	`“tag”: {“tagType”: “custom tag type1”, “tagValue”: “custom tag value1”}`
	If you omit the key, then enter just the value without a keyword for its field. Example:
	`“tag”: “custom tag value1”`

Field	Description
`tagType`	The key for a user-defined tag
`tagValue`	The value of the key for a user-defined tag
`deviceidlist`	(Required) An array of one or more device IDs to which you are applying the user-defined tag. Each item in the array is a string.

Request to Add a User-defined Tag as a Key-Value Pair Example

```
curl --location -X PUT 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/device/update?customerid=acmecorp' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY' \
--header 'Content-Type: application/json' \
--data-raw \
'{
    "tag":
      {
      "tagType": "Peninsula-Pacific", "tagValue": "F4"
      },
    "deviceidlist":
      [
      "00:e0:81:e6:01:4b",
      "00:e0:81:e6:02:55",
      "0c:c4:7a:a8:c3:22"
      ]
}'
```

Request to Add a User-defined Tag as a Value Example

```
curl --location -X PUT 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/device/update?customerid=acmecorp' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY' \
--header 'Content-Type: application/json' \
--data-raw \
'{
    "tag": "F4"
    "deviceidlist":
      [
      "00:e0:81:e6:01:4b",
      "00:e0:81:e6:02:55",
      "0c:c4:7a:a8:c3:22"
      ]
}'
```

Success Response

Upon success, the HTTP response code is 200.

Success Response Example

```
{
    "api": "/pub/v4.0/device/update",
    "ver":"v4.0",
    "code": 1,
    "message": "OK",
    "updatedDeviceNum": 3
}
```

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains invalid JSON in its body.
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests to add a user-defined tag exceeded the rate limit of 180 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```

Get a List of User-defined Tags

Synopsis

URI	/pub/v4.0/tag/list
HTTP Method	GET
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Get a list of user-defined tags assigned to IoT devices.

Request Fields

The URL of this request contains the following parameters:

Field	Description
`source`	(Required) This is the source of the user-defined tags assigned to IoT devices and must be followed by the string **`tenant`**.
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.

Request to Get a List of Tags Example

```
curl 'https://acmecorp.iot.paloaltonetworks.com/pub/v4.0/tag/list?
source=tenant&customerid=acmecorp' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

Success Response

Upon success, the HTTP response code is 200.

Success Response Example

```
{
    "ver": "0.3"
    "api": "tagRule",
    "totalTags": 119,
    "tags": [
      {
        "_id": {
            "tagId": "61d0cee45141f70700eb1612"
        },
        "createDate": "2022-01-01T22:00:04.569Z",
        "filters": [],
        "tagId": "61d0cee45141f70700eb1612",
        "tagType": "Owner",
        "tagValue": "Joe",
        "tenantid": "8181175672931450770",
        "type": "custom"
      }
      …
    ]
}
```

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains an invalid query string.
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests for a list of user-defined tags exceeded the rate limit of 180 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```

Get Active Policy Rule Recommendations

Synopsis

URI	/pub/v4.0/policy/recommendation
HTTP Method	GET
FQDN	<customer-name>.iot.paloaltonetworks.com

Description

Get a list of all active policy rule recommendations or all the active recommendations for one or more IoT device profiles.

Request Fields

The URL of this request contains the following parameters:

Field	Description
`customerid`	(Required) The customer ID specifies the API call for a specific tenant. The following value is a string.
`profile`	A profile filters policy rule recommendations by one or more source profile names. The following value is a string with profile names separated by commas; for example: `profile=Palo Alto Networks Device,iPhone,Polycom IP Phone`. All profiles must be IoT profiles. Without a profile filter, the request returns all active policy rule recommendations.

For other parameters you can include in the URL—`offset` and `pagelength`—see the general parameters described in Get Started with the IoT Security API.

Policy Rule Recommendations Request Example

```
curl --location -X GET 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/policy/recommendation?customerid=acmecorp' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

Success Response

Upon success, the HTTP response code is 200. In addition, this API returns a JSON object containing policy rules and their attributes.

An empty field indicates `any`. For example, if there are no IP addresses for `destinationIpList`, then the rule applies to any destination IP address.

Field	Description
`ver`	API version (string)
`api`	API path (string)
`total`	Total number of active recommended policy rules for which information was returned (integer)
`policies`	Array of active recommended policy rules (array)
`id`	Unique identifier composed of alphanumeric characters for the policy rule (string)
`policySetName`	Name of the user-defined policy set to which the policy rule belongs (string)
`geo`	Location of the destination in the policy rule (string): **`intranet`** (internal) or **`internet`** (external)
`action`	Action the firewall takes when applying the policy rule, which is always `allow` (string)
`lastActivityTime`	UTC timestamp for the last detected network activity corresponding to the elements in this policy rule (string)
`sourceProfiles`	Device profile assigned to devices initiating traffic to which the policy rule applies (array) *Although this is an array, there can only be a single source profile.*
`apps`	Applications to which the policy rule applies such as `youtube-base` (array)
`destinationProfiles`	Device profile of the destination in the policy rule. A destination device profile is used when the source and destination are in the

Field	Description
	same intranet and IoT Security is monitoring them both and has assigned a profile to the destination. (array)
`sourceIpList`	List of source IP addresses to which the policy rule applies (array) *This is included in anticipation of future functionality and is currently always empty.*
`destinationIpList`	List of destination IP addresses to which the policy rule applies (array) *When a destination is internal, IoT Security displays its IP address in* `destinationIpList`*. When it's external, IoT Security displays it in* `destinationFqdnList`*.*
`destinationFqdnList`	List of destination FQDNs to which the policy rule applies (array) *When a destination is external, IoT Security displays its IP address in* `destinationFqdnList`*. When it's internal, IoT Security displays it in* `destinationIpList`*.*
`sourceZones`	List of source zones to which the policy rule applies (array)
`destinationZones`	List of destination zones to which the policy rule applies (array)
`destinationUrlCategories`	List of categories to which the policy rule applies. Some examples: `games`, `entertainment`, and `health-and-medicine` (array)
`services`	List of non-standard service port numbers for an application or the user-defined values **`service-http`** and **`service-https`** (array)

Field	Description
	When IoT Security identifies an application that's using non-standard UDP or TCP port numbers, it displays the application name in "apps" and the non-standard port numbers in "services". When an application is using standard ports, IoT Security displays the application name and leaves "services" empty. If a user manually applied one of the predefined services `service-http` *or* `service-https` *to an application, then the predefined service name appears in "services".*
`tags`	System-defined tag `IoTSecurityRecommended` and any user-defined tags applied to the policy rule (array)
`securityProfiles`	List of Security profiles for antivirus, vulnerability protection, anti-spyware, and so on in the policy rule (array)
`firewallList`	List of firewalls that enforce the policy rule (array)
`deviceGroups`	(Panorama) List of device groups containing firewalls that enforce the policy rule (array)

Success Response Example

```
{
    "ver": "v4.0",
    "api": "/policy/recommendation",
    "total": 116,
    "policies": [
        {
            "id": "96122896cb71f1c302253842e1fb3518",
            "geo": "internet",
            "action": "allow",
            "lastActivityTime": "2021-06-03T04:43:26.400Z",
            "sourceProfiles": [
                "DICOM-Imager"
            ],
            "apps": [
                "cfdp"
            ],
            "destinationProfiles": [],
            "sourceIpList": [],
            "destinationIpList": [],
            "destinationFqdnList": [],
            "sourceZones": [],
            "destinationZones": [],
            "destinationUrlCategories": [],
            "services": [],
            "tags": [
                "IoTSecurityRecommended"
            ],
            "securityProfiles": [],
            "firewallList": [],
            "deviceGroups": []
        },
        ...
    ]
}
```

Policy Rule Recommendations for a Specific Profile Request Example

```
curl --location -X GET 'https://acmecorp.iot.paloaltonetworks.com/
pub/v4.0/policy/recommendation?customerid=acmecorp&profile=DICOM-
Imager' \
-H 'X-Key-Id: KEY_ID' \
-H 'X-Access-Key: ACCESS_KEY'
```

Error Response

Upon error, the reply includes an HTTP response code, an error message, and additional information describing the error. The HTTP response code is one of the following:

Field	Description
400	Bad Request. This occurs when an HTTP request contains an invalid query string.
403	Forbidden access. Either the provided API Key is invalid or it does not have the required RBAC permissions to run this API.
429	Too many requests. The number of requests for the list of recommended policy rules exceeded the rate limit of 180 queries per minute per tenant.
500	Internal server error. A unified status for API communication type errors.

Error Response Format

```
{code: STATUS_CODE, msg: GENERAL_MESSAGE}
```