



Monitoring Cisco Meraki (API)

Cisco: Meraki [API] PowerPack version 109

Table of Contents

Introduction	3
What is Cisco Meraki?	3
What Does the Cisco: Meraki [API] PowerPack Monitor?	4
Installing the Cisco: Meraki [API] PowerPack	4
Configuration and Discovery	6
Generating a Cisco Meraki API Key	7
Creating a Basic/Snippet Credential	9
Creating a Basic/Snippet Credential in the SL1 Classic User Interface	11
Creating a SOAP/XML Credential	12
Creating a SOAP/XML Credential in the SL1 Classic User Interface	15
Testing the Cisco Meraki API Credential	18
Testing the Cisco Meraki API Credential in the SL1 Classic User Interface	20
Disabling Asynchronous Dynamic Application Collection	22
Re-enabling Asynchronous Dynamic Application Collection	22
Creating a Cisco Meraki Virtual Device	24
Manually Aligning the Cisco: Meraki Cloud Controller Discovery Dynamic Application	24
Manually Aligning the Cisco: Meraki Cloud Controller Discovery Dynamic Application in the SL1 Classic User Interface	25
Configuring Dynamic Applications to Hide Empty Rows	25
Viewing Cisco Meraki Component Devices	26
Creating Events from Cisco Meraki Emails	28
Formatting Inbound Emails	28
Enabling Inbound Email Alerts	29
Adding Custom Device Classes to the PowerPack	30

Chapter

1

Introduction

Overview

This manual describes how to monitor Cisco Meraki access points, switches, phones, and cameras in SL1 using the *Cisco: Meraki [API] PowerPack* and the Meraki API.

The following sections provide an overview of Cisco Meraki and the *Cisco: Meraki [API] PowerPack*:

What is Cisco Meraki?	3
What Does the Cisco: Meraki [API] PowerPack Monitor?	4
Installing the Cisco: Meraki [API] PowerPack	4

NOTE: ScienceLogic provides this documentation for the convenience of ScienceLogic customers. Some of the configuration information contained herein pertains to third-party vendor software that is subject to change without notice to ScienceLogic. ScienceLogic makes every attempt to maintain accurate technical information and cannot be held responsible for defects or changes in third-party vendor software. There is no written or implied guarantee that information contained herein will work for all third-party variants. See the End User License Agreement (EULA) for more information.

What is Cisco Meraki?

Cisco Meraki provides a set of networking devices and appliances that you can manage from the cloud. Cisco Meraki's centralized cloud architecture enables you to securely monitor users, applications, and devices in your environment.

What Does the Cisco: Meraki [API] PowerPack Monitor?

To monitor Cisco Meraki devices using SL1 and the Meraki API, you must install the *Cisco: Meraki [API]* PowerPack. This PowerPack enables you to discover and collect data about Cisco Meraki appliances.

The *Cisco: Meraki [API]* PowerPack includes:

- Dynamic Applications to discover, model, and monitor performance metrics and collect configuration data for Cisco Meraki devices
- Device Classes for each of the Cisco Meraki devices that SL1 monitors
- Event Policies and corresponding alerts that are triggered when Cisco Meraki devices meet certain status criteria
- Example credentials that you can use as template to create Basic/Snippet or SOAP/XML credentials for connecting to the Cisco Meraki API
- Run Book Action and Automation policies that create a Meraki Cloud Controller virtual device during discovery and vanish devices

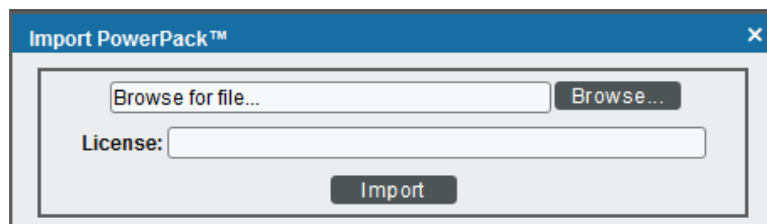
Installing the Cisco: Meraki [API] PowerPack

Before completing the steps in this manual, you must import and install the latest version of the *Cisco: Meraki [API]* PowerPack.

TIP: By default, installing a new version of a PowerPack overwrites all content from a previous version of that PowerPack that has already been installed on the target system. You can use the **Enable Selective PowerPack Field Protection** setting in the **Behavior Settings** page (System > Settings > Behavior) to prevent new PowerPacks from overwriting local changes for some commonly customized fields. (For more information, see the **System Administration** manual.)

To download and install a PowerPack:

1. Download the PowerPack from the [ScienceLogic Support Site](#).
2. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
3. In the **PowerPack Manager** page, click the **[Actions]** button, then select *Import PowerPack*.
4. The **Import PowerPack** dialog box appears:



5. Click the **[Browse]** button and navigate to the PowerPack file.
6. When the **PowerPack Installer** modal appears, click the **[Install]** button to install the PowerPack.

NOTE: If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPack Manager** page. However, the imported PowerPack will appear in the **Imported PowerPacks** modal. This page appears when you click the **[Actions]** menu and select *Install PowerPack*.

Chapter

2

Configuration and Discovery

Overview

The following sections describe how to configure and discover Cisco Meraki devices for monitoring by SL1 using the *Cisco: Meraki [API]* PowerPack and the Meraki API:

Generating a Cisco Meraki API Key	7
Creating a Basic/Snippet Credential	9
<i>Creating a Basic/Snippet Credential in the SL1 Classic User Interface</i>	11
Creating a SOAP/XML Credential	12
<i>Creating a SOAP/XML Credential in the SL1 Classic User Interface</i>	15
Testing the Cisco Meraki API Credential	18
<i>Testing the Cisco Meraki API Credential in the SL1 Classic User Interface</i>	20
Disabling Asynchronous Dynamic Application Collection	22
<i>Re-enabling Asynchronous Dynamic Application Collection</i>	22
Creating a Cisco Meraki Virtual Device	24
Manually Aligning the Cisco: Meraki Cloud Controller Discovery Dynamic Application	24
<i>Manually Aligning the Cisco: Meraki Cloud Controller Discovery Dynamic Application in the SL1 Classic User Interface</i>	25
<i>Configuring Dynamic Applications to Hide Empty Rows</i>	25
Viewing Cisco Meraki Component Devices	26
Creating Events from Cisco Meraki Emails	28
<i>Formatting Inbound Emails</i>	28
<i>Enabling Inbound Email Alerts</i>	29

Generating a Cisco Meraki API Key

To configure Cisco Meraki for monitoring using the Meraki API, you must first generate an API key for a read-only Meraki user. You will then enter this user's API key in the *Basic/Snippet credential* and *SOAP/XML credential* you create in SL1 to monitor Meraki.

NOTE: If the read-only user has access to multiple organizations, then SL1 can discover all of those organizations with a single discovery session. In this scenario, each organization is created as a separate Cloud Controller in SL1.

However, if you want each Meraki organization to have its own corresponding ScienceLogic organization in SL1, ScienceLogic recommends creating a unique read-only user account and API key for each organization in Meraki. You can then create separate credentials in SL1 for each Meraki organization using those unique API keys, and then use those credentials to run separate discovery sessions for each organization.

To create a read-only user:

1. Log in to the Cisco Meraki web interface.
2. Go to **Organization > Administrators**, and then click the **[Add admin]** button.
3. On the **Create administrator** page, complete the following fields:

The screenshot shows a 'Create administrator' modal window. It contains the following elements:

- Name:** A text input field.
- Email:** A text input field.
- Organization access:** A dropdown menu currently set to 'Read-only'.
- Table:** A table with two columns: 'Target' and 'Access'. Below the table is a green link that says '+ Add access privileges'.
- Footer:** A 'privacy' link on the left, and 'Close' and 'Create admin' buttons on the right.

- **Name.** Type the user's name.
- **Email.** Type the user's email address.
- **Organization access.** Select *Read-only*.

4. Click [**Create admin**]. Cisco Meraki sends an email to the email address provided, describing how the user can complete the registration process. The user must complete those steps before generating the API key.

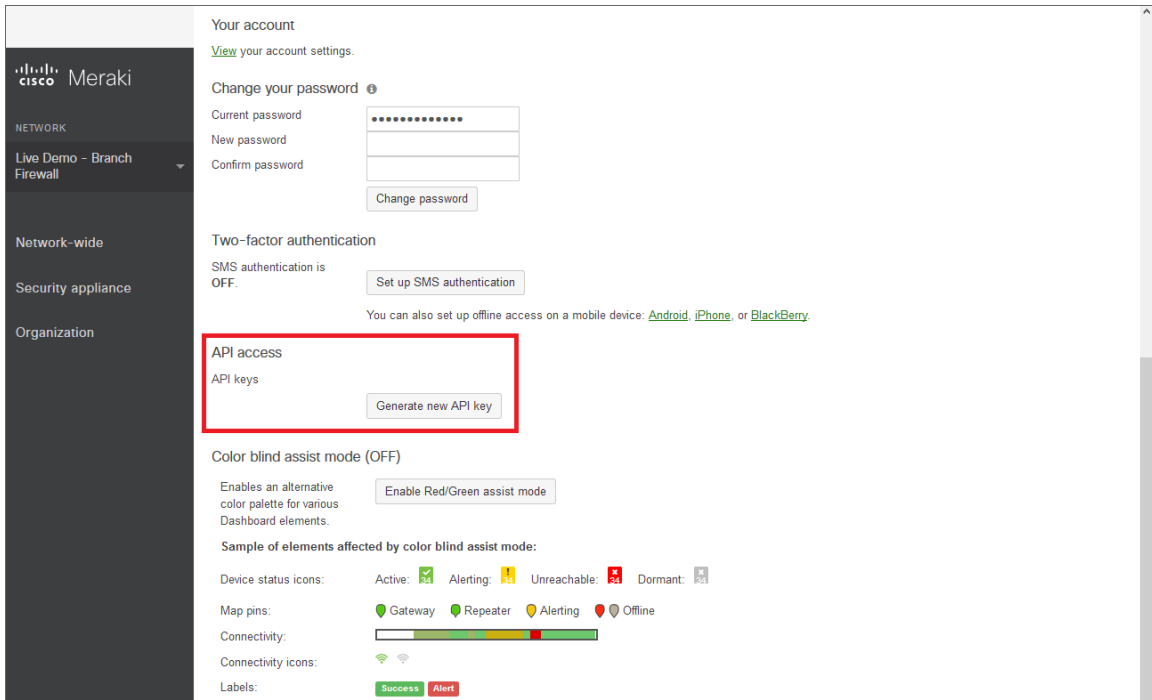
To generate a Cisco Meraki API key for that read-only user:

1. Log in to the Cisco Meraki web interface as the read-only user.
2. Go to **Organization > Settings**:



3. In the **Dashboard API access** section, select the **Enable access to the Cisco Meraki Dashboard API** checkbox.
4. Click the **Save Changes** button.
5. Click the **profile** link in the **Dashboard API access** section.

6. In your user profile, navigate to the **API access** section and click the **Generate new API key** button.



7. In the **API access** section, the API key appears. Copy and save the key value.

NOTE: API keys are visible only to the user that created them.

Creating a Basic/Snippet Credential

To configure SL1 to monitor Cisco Meraki systems using the Meraki API, you must create a Basic/Snippet credential. This credential allows the Dynamic Applications in the *Cisco: Meraki [API] PowerPack* to connect with the Cisco Meraki API. An example Basic/Snippet credential that you can edit for your own use is included in the PowerPack.

NOTE: If you are using an SL1 system prior to version 11.1.0, the new user interface does not include the **Duplicate** option for sample credential(s). ScienceLogic recommends that you use **the classic user interface and the Save As button** to create new credentials from sample credentials. This will prevent you from overwriting the sample credential(s).

To create a Basic/Snippet credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the **Cisco: Meraki - API** sample credential, click its **[Actions]** icon (⋮) and select **Duplicate**. A copy of the credential, called **Cisco: Meraki - API copy** appears.
3. Click the **[Actions]** icon (⋮) for the **Cisco: Meraki - API copy** credential and select **Edit**. The **Edit Credential** page appears:

4. Supply values in the following fields:
 - **Name**. Type a new name for the Meraki credential.
 - **All Organizations**. Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the **What organization manages this service?** drop-down field to align the credential with those specific organizations.
 - **Timeout (ms)**. Keep the default value.
 - **Hostname/IP**. Keep the default value.

NOTE: You **must** use the default value in the **Hostname/IP** field.

- **Port**. Keep the default value.
 - **Username**. Keep the default value.
 - **Password**. Type the **Meraki API key**.
4. Click **[Save & Close]**.

NOTE: If you would like to test your credential using the Credential Tester panel, click **[Save & Test]**. For detailed instructions on using the Credential Tester panel, see the [Testing the Cisco Meraki API Credential](#) section.

Creating a Basic/Snippet Credential in the SL1 Classic User Interface

To configure SL1 to monitor Cisco Meraki systems using the Meraki API, you must create a Basic/Snippet credential. This credential allows the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack to connect with the Cisco Meraki API. An example Basic/Snippet credential that you can edit for your own use is included in the PowerPack.

To create a Basic/Snippet credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the **Cisco: Meraki - API** credential, and then click its wrench icon (🔧). The **Edit Basic/Snippet Credential** modal page appears:

The screenshot shows a modal window titled "Credential Editor [130]" with a subtitle "Edit Basic/Snippet Credential #130". The window contains a "Basic Settings" section with the following fields:

- Credential Name:** Cisco: Meraki - API
- Hostname/IP:** https://api.meraki.com
- Port:** 443
- Timeout(ms):** 5000
- Username:** X-Cisco-Meraki-API-Key
- Password:** Masked with dots

Buttons include "New", "Reset", "Save", and "Save As".

3. Complete the following fields:
 - **Credential Name.** Type a new name for the credential.
 - **Hostname/IP.** Keep the default value.

NOTE: You **must** use the default value in the **Hostname/IP** field.

- **Port.** Keep the default value.
- **Timeout(ms).** Keep the default value.
- **Username.** Keep the default value.
- **Password.** Type the [Meraki API key](#).

4. Click the **[Save As]** button.
5. When the confirmation message appears, click **[OK]**.

Creating a SOAP/XML Credential

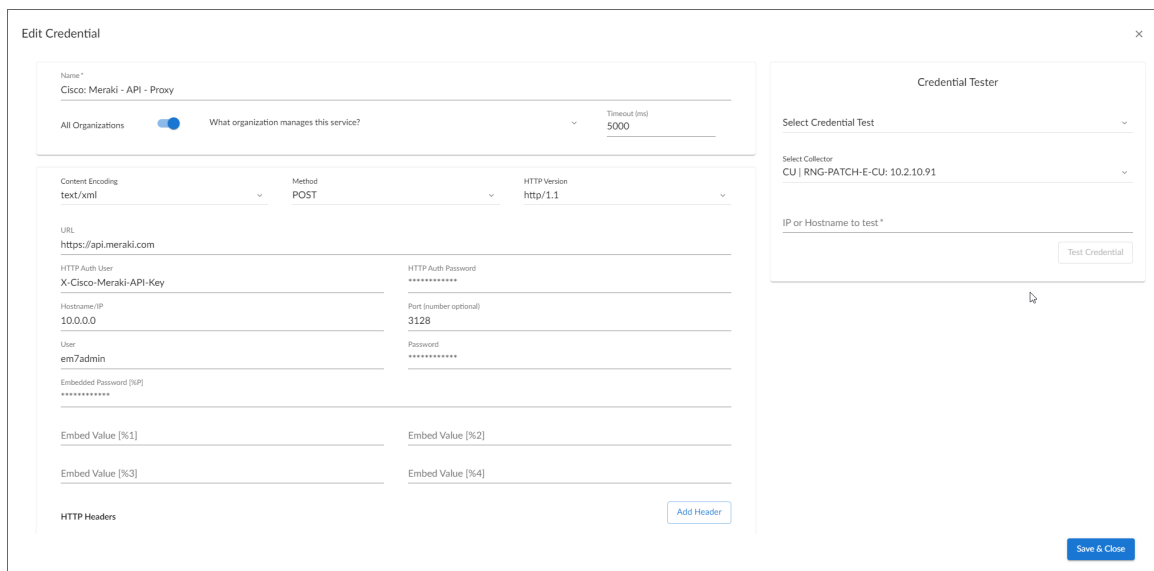
If you access Meraki systems through a third-party proxy server, you can create a SOAP/XML credential to enable the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack to connect with the Cisco Meraki API via the proxy server.

Similarly, if you want to discover only some selected devices, you can create a SOAP/XML credential that specifies tag values that the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack can use to determine which devices should be discovered.

NOTE: If you are on an SL1 system prior to version 11.1.0, you will not be able to duplicate the sample credential. It is recommended that you create your new credentials using [the SL1 classic user interface](#) so you do not overwrite the sample credential(s).

Two example SOAP/XML credentials that you can edit for your own use are included in the PowerPack:

- **Cisco: Meraki - API - Proxy**, for users who connect to Meraki through a third-party proxy server



- **Cisco: Meraki - API (Selective)**, for users who want to discover only some selected devices based on tag values

To define a SOAP/XML credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the sample credential you want to use, then click its **[Actions]** icon (**...**) and select **Duplicate**. A copy of the credential, called **Cisco: Meraki - API - Proxy copy** or **Cisco: Meraki - API (Selective) copy** appears.
3. Click the **[Actions]** icon (**...**) for the credential copy and select **Edit**. The **Edit Credential** modal page appears.

3. Supply values in the following fields:
 - **Name**. Type a new name for your Meraki credential.

- **All Organizations.** Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the **What organization manages this service?** drop-down field to align the credential with those specific organizations.
- **Timeout (ms).** Keep the default value.
- **Content Encoding.** Keep the default value.
- **Method.** Keep the default value.
- **HTTP Version.** Keep the default value.
- **URL.** Keep the default value of "https://api.meraki.com".
- **HTTP Auth User.** Keep the default value.
- **HTTP Auth Password.** Type the [Meraki API key](#).

Proxy Settings

NOTE: You must complete the **Proxy Settings** fields only if you connect to the Meraki API through a third-party proxy server. If you do not use a proxy to connect to Meraki, then you can leave these fields blank.

- **Hostname/IP.** Type the server's hostname or IP address.
- **Port.** Type the port on the proxy server to which you will connect.
- **User.** Type the username used to access the proxy server.
- **Password.** Type the password used to access the proxy server.

HTTP Headers

- **proxy_url_protocol:http.** Edit this header if you want to connect a proxy using a different protocol, such as http or https. The default value is "http".
- **Add a header.** Click **[Add a header]** once if you want to include tag values for SL1 to match when it discovers Meraki devices, or click **[Add a header]** twice if you want to include tag values and specify that tag-matching should be case-insensitive. In the blank fields that appear, do one or both of the following:
 - Type "tags:" in the first field, followed by one or more tag values. You can include multiple tag values in a string, using comma separators and no spaces. For example: "tags:value1,value2,value3".
 - Type "regex:IGNORECASE" in the second field if you want SL1 to match the tag values regardless of case.

NOTE: If you are using a tag to discover a device and want to discover that device's network, the device and its network must have the same tag applied.

NOTE: Tag values can include wildcard characters.

NOTE: After initial discovery, you can add more tag values and run discovery again to discover additional component devices. However, if you remove tag values and then run discovery again, the component devices that had been discovered based on the removed tag values will be updated to an unavailable state.

4. Click **[Save & Close]**.

NOTE: If you would like to test your credential using the Credential Tester panel, click **[Save & Test]**. For detailed instructions on using the Credential Tester panel, see the [Testing the Cisco Meraki API Credential](#) section.

Creating a SOAP/XML Credential in the SL1 Classic User Interface

If you access Meraki systems through a third-party proxy server, you can create a SOAP/XML credential to enable the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack to connect with the Cisco Meraki API via the proxy server.

Similarly, if you want to discover only some selected devices, you can create a SOAP/XML credential that specifies tag values that the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack can use to determine which devices should be discovered.

Two example SOAP/XML credentials that you can edit for your own use are included in the PowerPack:

- **Cisco: Meraki - API - Proxy**, for users who connect to Meraki through a third-party proxy server

Credential Editor [87]

Edit SOAP/XML Credential #87 [New] [Reset]

Basic Settings

Profile Name: Cisco: Meraki - API - Proxy
 Content Encoding: [text/xml]
 Method: [POST]
 HTTP Version: [HTTP/1.1]

URL [http(s)://Host:Port/Path | %D = Aligned Device Address | %N = Aligned Device Host Name]
 https://api.meraki.com

HTTP Auth User: X-Cisco-Meraki-API-Key
 HTTP Auth Password: [REDACTED]
 Timeout (seconds): 5

Proxy Settings

Hostname/IP: 10.0.0.0
 Port: 3128
 User: em7admin

CURL Options

CAINFO
 CAPATH
 CLOSEPOLICY
 CONNECTTIMEOUT
 COOKIE
 COOKIEFILE
 COOKIEJAR
 COOKIELIST
 CRLF
 CUSTOMREQUEST
 DNSCACHETIMEOUT

Soap Options

Embedded Password [%P]
 Embed Value [%1]
 Embed Value [%2]
 Embed Value [%3]
 Embed Value [%4]

HTTP Headers

+ Add a header
 proxy_url_protocol:http

[Save] [Save As]

- **Cisco: Meraki - API (Selective)**, for users who want to discover only some selected devices based on tag values

Credential Editor [88]

Edit SOAP/XML Credential #88 [New] [Reset]

Basic Settings

Profile Name: Cisco: Meraki - API (Selective)
 Content Encoding: [text/xml]
 Method: [POST]
 HTTP Version: [HTTP/1.1]

URL [http(s)://Host:Port/Path | %D = Aligned Device Address | %N = Aligned Device Host Name]
 https://api.meraki.com

HTTP Auth User: X-Cisco-Meraki-API-Key
 HTTP Auth Password: [REDACTED]
 Timeout (seconds): 5

Proxy Settings

Hostname/IP: [REDACTED]
 Port: 0
 User: [REDACTED]

CURL Options

CAINFO
 CAPATH
 CLOSEPOLICY
 CONNECTTIMEOUT
 COOKIE
 COOKIEFILE
 COOKIEJAR
 COOKIELIST
 CRLF
 CUSTOMREQUEST
 DNSCACHETIMEOUT

Soap Options

Embedded Password [%P]
 Embed Value [%1]
 Embed Value [%2]
 Embed Value [%3]
 Embed Value [%4]

HTTP Headers

+ Add a header
 tags:Science?gic
 regex:IGNORECASE

[Save] [Save As]

To define a SOAP/XML credential:

1. Go to the **Credential Management** page (System > Manage > Credentials).

2. Locate the Cisco Meraki example credential that you want to use and click its wrench icon (🔧). The **Credential Editor** modal page appears:

Credential Editor [88]

Edit SOAP/XML Credential #88 New Reset

Basic Settings

Profile Name: Cisco: Meraki - API (Selective) | Content Encoding: [text/xml] | Method: [POST] | HTTP Version: [HTTP/1.1]

URL [http(s)://Host:Port/Path | %D = Aligned Device Address | %N = Aligned Device Host Name]
https://api.meraki.com

HTTP Auth User: X-Cisco-Meraki-API-Key | HTTP Auth Password: | Timeout (seconds): 5

Proxy Settings

Hostname/IP: | Port: 0 | User: |

CURL Options

CAINFO, CAPATH, CLOSEPOLICY, CONNECTTIMEOUT, COOKIE, COOKIEFILE, COOKIEJAR, COOKIELIST, CRLF, CUSTOMREQUEST, DNSCACHETIMEOUT

Soap Options

Embedded Password [%P]

Embed Value [%1] | Embed Value [%2]

Embed Value [%3] | Embed Value [%4]

HTTP Headers

+ Add a header

tags:Science!?gic

regex:IGNORECASE

Save Save As

3. Enter values in the following fields:

Basic Settings

- **Profile Name.** Type a new name for your Meraki credential.
- **HTTP Auth Password.** Type the [Meraki API key](#).

NOTE: You can use the default values for the remaining **Basic Settings** fields. You **must** use the default value in the **URL** field.

Proxy Settings

NOTE: You must complete the **Proxy Settings** fields only if you connect to the Meraki API through a third-party proxy server. If you do not use a proxy to connect to Meraki, then you can leave these fields blank.

- **Hostname/IP.** Type the server's hostname or IP address.
- **Port.** Type the port on the proxy server to which you will connect.
- **User.** Type the username used to access the proxy server.

- **Password.** Type the password used to access the proxy server.

HTTP Headers

- **proxy_url_protocol: http.** Edit this header if you want to connect a proxy using a different protocol, such as http or https. The default value is "http".
- **Add a header.** Click **[Add a header]** once if you want to include tag values for SL1 to match when it discovers Meraki devices, or click **[Add a header]** twice if you want to include tag values and specify that tag-matching should be case-insensitive. In the blank fields that appear, do one or both of the following:
 - Type "tags:" in the first field, followed by one or more tag values. You can include multiple tag values in a string, using comma separators and no spaces. For example: "tags:value1,value2,value3".
 - Type "regex:IGNORECASE" in the second field if you want SL1 to match the tag values regardless of case.

NOTE: If you are using a tag to discover a device and want to discover that device's network, the device and its network must have the same tag applied.

NOTE: Tag values can include wildcard characters.

NOTE: After initial discovery, you can add more tag values and run discovery again to discover additional component devices. However, if you remove tag values and then run discovery again, the component devices that had been discovered based on the removed tag values will be updated to an unavailable state.

4. Click the **[Save As]** button, and then click **[OK]**.

Testing the Cisco Meraki API Credential

The *Cisco: Meraki [API]* PowerPack includes a credential test for Cisco Meraki credentials. Credential tests define a series of steps that SL1 can execute on demand to validate whether a credential works as expected.

The Cisco Meraki credential tests can be used to test the Basic/Snippet and SOAP/XML credentials for monitoring the Cisco Meraki API using the Dynamic Applications in the *Cisco: Meraki [API]* PowerPack.

The **Cisco: Meraki [API] (Basic/Snippet) Credential tester** performs the following steps:

- **Test Meraki Reachability.** Performs an ICMP ping request to the host specified in the credential.
- **Test Meraki Port Availability.** Performs an NMAP request to the TCP port specified in the credential on the host specified in the credential.

- **Test Meraki Organization Request.** Performs a check to see if the Meraki organization request has been collected appropriately.

The **Cisco: Meraki [API] (SOAP/XML) Credential tester** performs the following steps:

- **Test Meraki Reachability.** Performs an ICMP ping request to the host specified in the credential.
- **Test Meraki Port Availability.** Performs an NMAP request to the TCP port specified in the credential on the host specified in the credential.
- **Test Meraki Organization Request.** Performs a check to see if the Meraki organization request has been collected appropriately.

To test the Cisco Meraki credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the credential you wish to test, select the **Actions** button (☰) next to it and click *Edit/Test*. The **Edit Credential** modal page appears:

3. In the **Credential Tester** pane on the right, fill out the following fields on this page:
 - **Select Credential Test.** Select **Cisco: Meraki [API] (Basic/Snippet) Credential tester** or the **Cisco: Meraki [API] (SOAP/XML) Credential tester**, depending on which credential you are testing.
 - **Collector.** Select the All-In-One Appliance or Data Collector that will run the test.
 - **IP or Hostname to Test.** Enter "api.meraki.com" or the IP address of your Meraki system.
4. Click the **[Run Test]** button to run the credential test. The **Testing Credential** window appears.

STEP	DESCRIPTION	LOG MESSAGE	STATUS
Test Meraki reachability.	Check to see if the IP/Hostname is reachable using ICMP.	The state of IP/Hostname [api.meraki.com] is 'reachable' using ICMP, t...	Passed
Test Meraki port availa...	Check to see if the Meraki port has been open appropriately.	The IP/Hostname [api.meraki.com] is using the port [443], the current...	Passed
Test Meraki organizati...	Check to see if the Meraki organizations request has been collected a...	Collected: 4 Organizations.	Passed

The **Testing Credential** window displays a log entry for each step in the credential test. The steps performed are different for each credential test. The log entry for each step includes the following information:

- **Step.** The name of the step.
- **Description.** A description of the action performed during the step.
- **Log Message.** The result of the step for this execution of the credential test.
- **Status.** Whether the result of this step indicates the credential and/or the network environment is configured correctly (Passed) or incorrectly (Failed).
- **Step Tip.** Mouse over the question mark icon (?) to display the tip text. The tip text recommends what to do to change the credential and/or the network environment if the step has a status of "Failed".

Testing the Cisco Meraki API Credential in the SL1 Classic User Interface

The *Cisco: Meraki [API]* PowerPack includes a credential test for Cisco Meraki credentials. Credential tests define a series of steps that SL1 can execute on demand to validate whether a credential works as expected.

The Cisco Meraki credential tests can be used to test the Basic/Snippet and SOAP/XML credentials for monitoring the Cisco Meraki API using the Dynamic Applications in the *Cisco: Meraki [API]*PowerPack.

The **Cisco: Meraki [API] (Basic/Snippet) Credential tester** performs the following steps:

- **Test Meraki Reachability.** Performs an ICMP ping request to the host specified in the credential.
- **Test Meraki Port Availability.** Performs an NMAP request to the TCP port specified in the credential on the host specified in the credential.
- **Test Meraki Organization Request.** Performs a check to see if the Meraki organization request has been collected appropriately.

The **Cisco: Meraki [API] (SOAP/XML) Credential tester** performs the following steps:

- **Test Meraki Reachability.** Performs an ICMP ping request to the host specified in the credential.
- **Test Meraki Port Availability.** Performs an NMAP request to the TCP port specified in the credential on the host specified in the credential.

- **Test Meraki Organization Request.** Performs a check to see if the Meraki organization request has been collected appropriately.

To test the Cisco Meraki credential:

1. Go to the **Credential Test Management** page (System > Customize > Credential Tests).
2. Locate the **Cisco: Meraki [API] (Basic/Snippet) Credential tester** or the **Cisco: Meraki [API] (SOAP/XML) Credential tester**, depending on which credential you are testing, and click its lightning bolt icon (⚡). The **Credential Tester** modal page appears:

3. Supply values in the following fields:
 - **Test Type.** This field is pre-populated with the credential test you selected.
 - **Credential.** Select the credential to test. This drop-down list includes only credentials that you have access to that can be tested using the selected credential test.
 - **Hostname/IP.** Enter "api.meraki.com" or the IP address of your Meraki system.
 - **Collector.** Select the All-In-One Appliance or Data Collector that will run the test.
4. Click the **[Run Test]** button to run the credential test. The **Test Credential** window appears.

Step	Description	Log Message	Status
1 Test Meraki reachability.	Check to see if the IP/Hostname is reachable using ICMP.	The state of IP/Hostname [api.meraki.com] is 'reachable' using ICMP, the average response in time is 39.421ms	Passed
2 Test Meraki port availability.	Check to see if the Meraki port has been open appropriately.	The IP/Hostname [api.meraki.com] is using the port [443], the current state is 'Open'.	Passed
3 Test Meraki organizations request.	Check to see if the Meraki organizations request has been collected appropriately.	Collected: 29 Organizations.	Passed

The **Test Credential** window displays a log entry for each step in the credential test. The steps performed are different for each credential test. The log entry for each step includes the following information:

- **Step.** The name of the step.
- **Description.** A description of the action performed during the step.
- **Log Message.** The result of the step for this credential test.
- **Status.** Whether the result of this step indicates the credential or the network environment is configured correctly (Passed) or incorrectly (Failed).

- **Step Tip.** Mouse over the question mark icon (?) to display the tip text. The tip text recommends what to do to change the credential or the network environment if the step has a status of "Failed".

Disabling Asynchronous Dynamic Application Collection

If the Meraki system you want to monitor consists of more than 200 devices, you must disable the "Data Collection: Async Dynamic App Collection" process before discovering your Meraki system.

NOTE: Disabling asynchronous Dynamic Application collection increases the amount of time it takes the ScienceLogic platform to discover all of the component devices in your Meraki system.

To disable asynchronous Dynamic Application collection:

1. Go to the **Process Manager** page (System > Settings > Admin Processes, or System > Settings > Processes in the SL1 classic user interface).
2. Use the **Process Name** filter field to search for the "Data Collection: Async Dynamic App Collection" process, and then click its wrench icon (?). The **Process Editor** page appears.

The screenshot shows the 'Process Editor' window for 'Data Collection: Async Dynamic App Collection'. The 'Operating State' dropdown is highlighted with a red box and set to 'Disabled'. Other settings include Frequency: [Asynchronous], Async Throttle: [2], Time Factor (Mins.): [15], and various Appliance Types checked.

3. In the **Operating State** field, select *Disabled*.
4. Click **[Save]**.

Re-enabling Asynchronous Dynamic Application Collection

If you no longer want to monitor Meraki devices in SL1 and you want to return the system to its original state with asynchronous Dynamic Application collection re-enabled, you must first delete all Meraki devices from the platform. You must then clear the Database Server or Data Collector of any asynchronous processes that are already queued. Failing to do these steps can result in the platform ceasing all data collection until those asynchronous processes are executed.

To re-enable asynchronous Dynamic Application collection:

1. Navigate to the Database Server by typing "<IP address>:8008" into your browser address bar.

2. Log in to the Database Server. The phpMyAdmin browser appears.
3. Select the database from the drop-down **Database** field, and then select the **master_logs** database.
4. In the **master_logs** database, select the **spool_process** table on the left menu, and then click the **[SQL]** tab.
5. Run the following query to clear out the processes on the database:

```
DELETE FROM 'spool_process' WHERE 'proc' = 129 AND 'state' != 0;
```



6. Click **[OK]** at the prompt. Many rows should have been deleted from the table.


If you are using a distributed ScienceLogic system, continue with step 7. Otherwise, go to step 14.

7. In the left menu of the phpMyAdmin browser, select the Data Collector appliance where Meraki devices were discovered.

If the IP address of the Data Collector appears in the upper left-hand corner of the phpMyAdmin browser, go to step 12. Otherwise, if you receive a MySQL error message that your access is denied, continue with step 8.

8. In the Database Server, navigate to the **Master** database and then select the **system_settings_licenses** table.
9. Click **[Browse]** in the upper left-hand side of the page and then identify the Data Collector appliance.
10. Click the **edit** button for the Data Collector:

<input type="checkbox"/>			3	5	SL_ISO1_CU	collector unit: 10.2.8.72	8.5.0	2119	80500002119
--------------------------	---	---	---	---	------------	---------------------------------	-------	------	-------------

11. Locate the **db_user** and **db_pass** fields. In those fields, type the same credentials as the Database Server.
12. Click **[Go]**. Wait a few seconds before trying to access the Data Collector in the phpMyAdmin browser. When you do so, the IP address of the Data Collector should appear in the upper left-hand corner of the phpMyAdmin browser.
13. Repeat steps 3-6 on the Data Collector. If successful, many rows should have been deleted from the **spool_process** table.
14. In SL1, go to the **Process Manager** page (System > Settings > Admin Processes, or System > Settings > Processes in the SL1 classic user interface).
15. Use the **Process Name** filter field to search for the "Data Collection: Async Dynamic App Collection" process, and then click its wrench icon (). The **Process Editor** page appears.
16. In the **Operating State** field, select *Enabled*, and then click **[Save]**.

Creating a Cisco Meraki Virtual Device

To monitor your Cisco Meraki devices, you must create a **virtual device** that represents the Meraki Cloud Controller. A virtual device is a user-defined container that represents a device or service that cannot be discovered by SL1. You can use the virtual device to store information gathered by policies or Dynamic Applications.

If you want to discover more than one Meraki account, you must create a virtual device for each API key that you want to use.

To create a virtual device that represents your Meraki Cloud Controller:

1. Go to the **Device Manager** page (Devices > Device Manager).
2. Click **[Actions]** and select *Create Virtual Device* from the menu. The **Virtual Device** modal page appears.
3. Enter values in the following fields:
 - **Device Name.** Enter a name for the device.
 - **Organization.** Select the organization for this device. The organization you associate with the device limits the users that will be able to view and edit the device. Typically, only members of the organization will be able to view and edit the device.
 - **Device Class.** Select *Cisco Systems | Meraki Cloud Controller*.
 - **Collector.** Select the collector group that will monitor the device.
4. Click **[Add]** to create the virtual device.
5. Repeat these steps for each Meraki API key that you want to use.

Manually Aligning the Cisco: Meraki Cloud Controller Discovery Dynamic Application

After creating the Cisco Meraki virtual device, you must manually align the "Cisco: Meraki Cloud Controller Discovery" Dynamic Application to the Cisco Meraki virtual device.

To manually align the Cisco Meraki Dynamic Application:

1. Go to the **Devices** page (Devices > Device Manager).
2. Locate your Cisco Meraki virtual device and click its name.
3. In the **Device Investigator**, click the **[Collections]** tab.
4. Click the **[Edit]** button at the top of the page, then click the **[Align Dynamic App]** button.
5. In the **Align Dynamic Application** modal, click **Choose Dynamic Application**.
6. Locate the "Cisco: Meraki Cloud Controller Discovery" and click **[Select]**.
7. In the **Align Dynamic Application** modal, de-select the **Use Device SNMP Credential** box. Click the **Choose Credential** option that appears.
8. Select the Cisco Meraki credential you created and click **[Select]**.

9. Click **[Align Dynamic App]** to align the Dynamic Application with the Cisco Meraki virtual device.


After aligning the "Cisco: Meraki Cloud Controller Discovery" Dynamic Application, your Cisco Meraki component devices will be discovered and classified.

NOTE: The **Poll Frequency** of the "Cisco: Meraki Cloud Controller Discovery" Dynamic Application should be set to 5 minutes.

Manually Aligning the Cisco: Meraki Cloud Controller Discovery Dynamic Application in the SL1 Classic User Interface

After creating the Cisco Meraki virtual device, you must manually align the "Cisco: Meraki Cloud Controller Discovery" Dynamic Application to the Cisco Meraki virtual device.

To manually align the Cisco Meraki Dynamic Application:

1. Go to the **Device Manager** page (Devices > Device Manager).
2. Click the wrench icon () for your Cisco Meraki virtual device.
3. In the **Device Administration** panel, click the **[Collections]** tab. The **Dynamic Application Collections** page appears.
4. Click the **[Actions]** button and select *Add Dynamic Application* from the menu.
5. In the **Dynamic Application Alignment** window, from the **Dynamic Applications** field, select the "Cisco: Meraki Cloud Controller Discovery" Dynamic Application.
6. In the **Credentials** field, select the Cisco Meraki credential you created.
7. Click **[Save]** to align the Dynamic Application with the Cisco Meraki virtual device.


After aligning the "Cisco: Meraki Cloud Controller Discovery" Dynamic Application, your Cisco Meraki component devices will be discovered and classified.

NOTE: The **Poll Frequency** of the "Cisco: Meraki Cloud Controller Discovery" Dynamic Application should be set to 5 minutes.

Configuring Dynamic Applications to Hide Empty Rows

If you have a device that is no longer being monitored and a configuration Dynamic Application is returning empty rows in the **[Configs]** tab of that device, you can use the *Hide row* setting in the Dynamic Applications to hide those empty rows.

To do this:

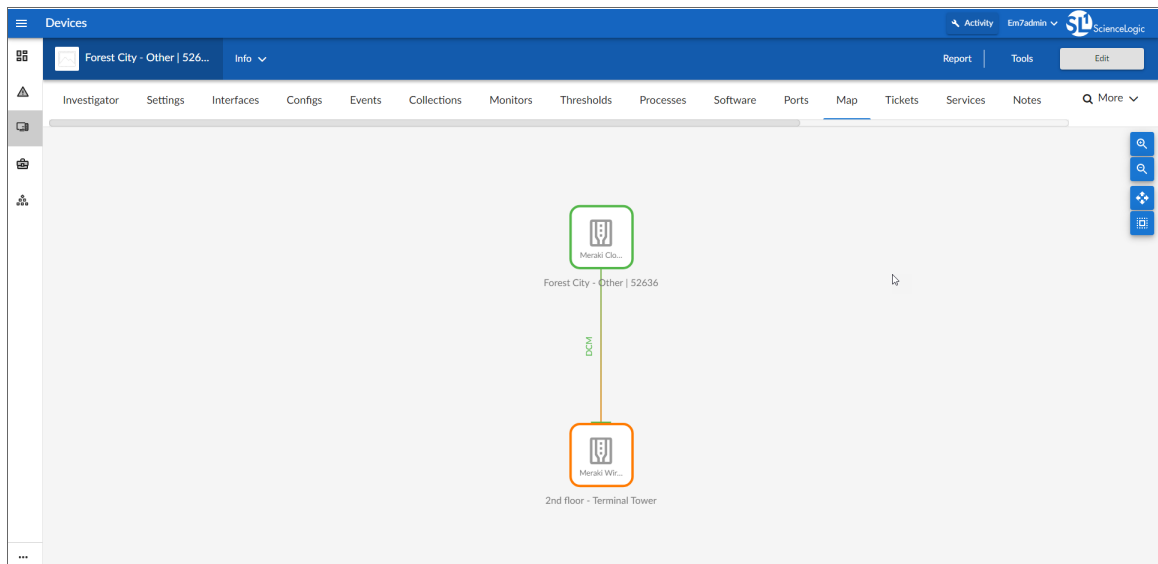
1. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
2. Located the *Cisco: Meraki [API]* PowerPack and click its wrench icon ()

3. In the pane on the left, select **Dynamic Applications**.
4. For each Dynamic Application with "Configuration" in its **Type**, click its wrench icon (🔧).
5. In the **Dynamic Applications Properties Editor**, click the **Null Row Option** dropdown and select *Hide row*.
6. Click **[Save]**.

Viewing Cisco Meraki Component Devices

In addition to the **Devices** page, you can view your Cisco Meraki devices in the following places in the user interface:

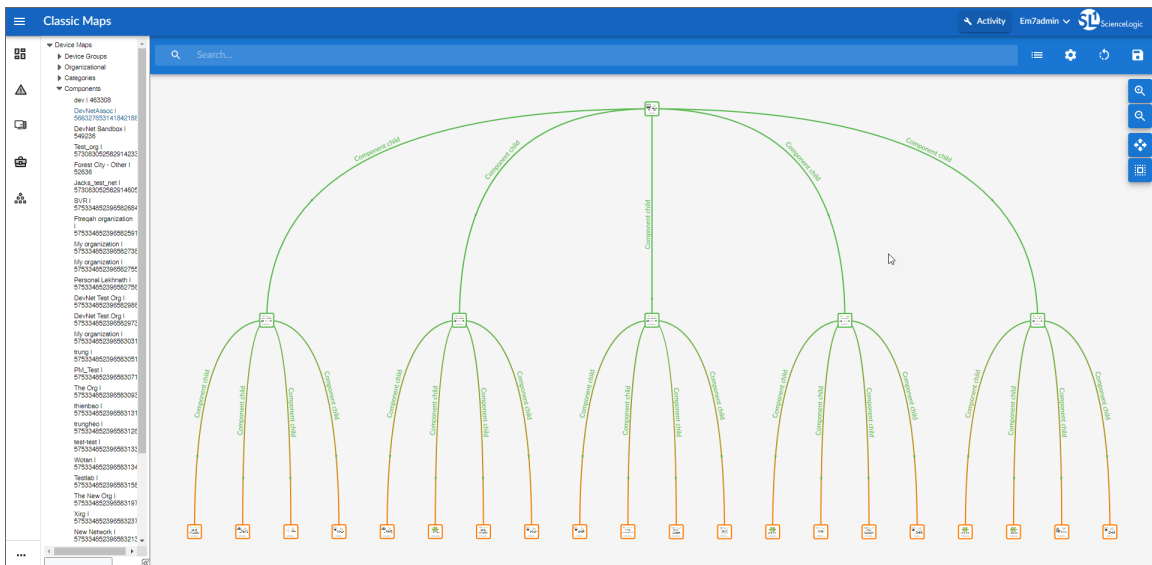
- The **Device Investigator** Map page (click **Map** in the **Device Investigator** page) displays a map of a particular device and all of the devices with which it has parent-child relationships. Double-clicking any of the listed devices reloads the page to make the selected device the primary device.



- The **Device Components** page (Devices > Device Components) displays a list of all root devices and component devices discovered by SL1. The **Device Components** page displays all root devices and component devices in an indented view, so you can easily view the hierarchy and relationships between child devices, parent devices, and root devices. To view the component devices associated with a Cisco Meraki device, find the device and click its plus icon (+).

Device Name	IP Address	Device Category	Device Class / Sub-class	DD	Organization	Current State	Collection Group	Collection State
Cisco Live US 2019 1 665776	--	Virtual	Cisco Systems Meraki Cloud Controller	3249	meraki_Sandbox	Healthy	CUIG1	Active
DeLab 1 661195	--	Virtual	Cisco Systems Meraki Cloud Controller	3248	meraki_Sandbox	Healthy	CUIG1	Active
DevNetLab	--	Network	Cisco Systems Meraki Combination Network	3281	meraki_Sandbox	Major	CUIG1	Unavailable
QDRP-EC97-MR89	--	Switches	Cisco Systems Meraki MS220-8 Series	3423	meraki_Sandbox	Unavailable	CUIG1	Unavailable
QGLD-FGMS-VP76	--	Access Point	Cisco Systems Meraki MR52	3422	meraki_Sandbox	Healthy	CUIG1	Unavailable
QZON-ORLY-NP7J	--	Firewall	Cisco Systems Meraki MX55	3421	meraki_Sandbox	Healthy	CUIG1	Unavailable
DevNetLab2	--	Network	Cisco Systems Meraki Combination Network	3275	meraki_Sandbox	Major	CUIG1	Unavailable
DevNetLab3	--	Network	Cisco Systems Meraki Combination Network	3273	meraki_Sandbox	Major	CUIG1	Unavailable
DNEAiershat	--	Network	Cisco Systems Meraki Combination Network	3271	meraki_Sandbox	Major	CUIG1	Unavailable
Lycel	--	Network	Cisco Systems Meraki Combination Network	3270	meraki_Sandbox	Major	CUIG1	Unavailable
Lycel MEM	--	Network	Cisco Systems Meraki EMM Network	3270	meraki_Sandbox	Major	CUIG1	Unavailable
Nolan	--	Network	Cisco Systems Meraki Wireless Network	3277	meraki_Sandbox	Major	CUIG1	Unavailable
Vegae Apartment	--	Network	Cisco Systems Meraki Combination Network	3269	meraki_Sandbox	Major	CUIG1	Unavailable
dev 1 463308	--	Virtual	Cisco Systems Meraki Cloud Controller	3221	meraki_Sandbox	Healthy	CUIG1	Active
DevNet Sandbox 1 662296	--	Virtual	Cisco Systems Meraki Cloud Controller	3226	meraki_Sandbox	Healthy	CUIG1	Active
DevNet Test Cfg 1 5753485259662973	--	Virtual	Cisco Systems Meraki Cloud Controller	3232	meraki_Sandbox	Healthy	CUIG1	Active
DevNet Test Cfg 1 5753485259662966	--	Virtual	Cisco Systems Meraki Cloud Controller	3234	meraki_Sandbox	Healthy	CUIG1	Active
DevNetAssoc 1 56627955141842188	--	Virtual	Cisco Systems Meraki Cloud Controller	3222	meraki_Sandbox	Healthy	CUIG1	Active
Forest City - Other 1 52696	--	Virtual	Cisco Systems Meraki Cloud Controller	3223	meraki_Sandbox	Healthy	CUIG1	Active
Firenash organization 1 5753485259662951	--	Virtual	Cisco Systems Meraki Cloud Controller	3227	meraki_Sandbox	Healthy	CUIG1	Active
Jacke_test_net 1 57534852596629466	--	Virtual	Cisco Systems Meraki Cloud Controller	3225	meraki_Sandbox	Healthy	CUIG1	Active
My organization 1 5753485259662938	--	Virtual	Cisco Systems Meraki Cloud Controller	3229	meraki_Sandbox	Healthy	CUIG1	Active

- The **Component Map** page (Classic Maps > Device Maps > Components) allows you to view devices by root node and view the relationships between root nodes, parent components, and child components in a map. This makes it easy to visualize and manage root nodes and their components. SL1 automatically updates the **Component Map** as new component devices are discovered. The platform also updates each map with the latest status and event information. To view the map for a Cisco Meraki device, go to the **Component Map** page and select the map from the list in the left NavBar. To learn more about the **Component Map** page, see the **Maps** manual.



Creating Events from Cisco Meraki Emails

The *Cisco: Meraki [API]* PowerPack includes Event Policies that can generate events in SL1 based on emails that Cisco Meraki sends to SL1.

For SL1 to process events from inbound emails, you must configure your Meraki devices to send email to SL1 using certain formatting rules.

You must then enable SL1 to generate events from those inbound Meraki emails.

If configured properly, when SL1 domain receives an email with body text that matches a Meraki network component device name and a subject that matches the regular expression (RegEx) pattern of one of the PowerPack's Event Policies, SL1 will generate an event aligned to that network component device.

NOTE: Events from email are always aligned to network devices, even when the email includes references to one or more sub-component devices below the network device.

CAUTION: The email Event Policies included in the *Cisco: Meraki [API]* PowerPack each have an expiry delay setting that specifies the amount of time after which an active event is automatically cleared from SL1 if the event has not reoccurred. However, SL1 clearing an event for reaching its expiry delay setting does not mean that the initial condition that caused the event has been resolved.

Formatting Inbound Emails

Inbound emails must meet the following requirements to be processed as events by SL1:

- The email must be sent to the following address:

`notify@SL1-domain-name`

Where "SL1-domain-name" is one of the fully qualified domain names of the Database Server or All-In-One Appliance that is entered in the **Authorized Email Domains** field in the **Email Settings** (System > Settings > Email) page.

- The "from" address used by the external device must be "alerts-noreply@meraki.com" for non-maintenance events, "support-noreply@meraki.com" for maintenance events, or otherwise match an address defined in the **Originator Address** field in an email redirection policy on the **Mailer Redirection** page (Events > Inbound Email, or Registry > Events > Inbound Email in the SL1 classic user interface).
- The email subject line must begin with "Alert for" or "Scheduled maintenance for" and match the regular expression (RegEx) pattern of one of the Event Policies included in the *Cisco: Meraki [API]* PowerPack.

- The email body must include the name of a network device monitored by the SL1 system.

The following RegEx patterns are used:

- For scheduled maintenance emails:

```
(Scheduled maintenance for)\s((network\s|\d\snetworks\s\sin\sorganization\s)"([a-zA-Z0-9_-\.\.]+).*)
```

- For all other emails:

```
(Alert for)\s*([a-zA-Z0-9_-\.\.]+)\s*
```

NOTE: There must be a space between the RegEx pattern and the IP address, hostname, or device ID.

NOTE: The Event Policies included in the *Cisco: Meraki [API] PowerPack* **do not** include RegEx patterns "out of the box". Users can add or modify Event Policy RegEx patterns to best suit their needs.

NOTE: Emails that do not match the RegEx pattern of any Meraki Event Policy will generate a message in the system log. Emails that do not match the name of any component device in SL1 will not generate any events or messages.

NOTE: You can specify how an Event from Email policy will match a RegEx to a device name in the **Behavior Settings** page (System > Settings > Behavior). For more information, see the *Configuring Inbound Email* manual.

Enabling Inbound Email Alerts

After you have ensured that inbound Meraki emails are formatted correctly, you must enable SL1 to generate events from the inbound Meraki emails.

To do so:

1. Go to the **Emailer Redirection** page (Events > Inbound Email, or Registry > Events > Inbound Email in the SL1 classic user interface), and then click the **[Create]** button. The **Add Policy** modal page appears.

2. Complete the following fields:

The screenshot shows a configuration form titled "Add Policy | Create New" with a "Reset" button in the top right corner. The form contains the following fields:

- Originator Address:** A text input field containing "alerts-noreply@meraki.com".
- Alignment Type:** A dropdown menu with the selected option "[If device not found, discard unmatched email]".
- Regex Pattern:** A text input field containing "Alert for".
- Regex Pattern Type:** A dropdown menu with the selected option "Advanced".
- Regex Type:** A dropdown menu with the selected option "[Subject]".

A "Save" button is located at the bottom center of the form.

- **Originator Address.** Type "alerts-noreply@meraki.com".
- **Alignment Type.** Select *If device not found, discard unmatched email*.
- **Regex Pattern.** Type "Alert for" or "Scheduled maintenance for network".
- **Regex Pattern Type.** Select *Advanced*.
- **Regex Type.** Select *Subject*.

3. Click [**Save**].

NOTE: For more information about generating events from inbound emails, see the *Configuring Inbound Email* manual.

Adding Custom Device Classes to the PowerPack

If you have created custom device classes for your Cisco Meraki devices, you can add them to the PowerPack.

For more information on how to create device classes, see the *Device Management* manual.

To add device classes to the PowerPack:

1. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
2. Locate the *Cisco: Meraki [API]* PowerPack and click its wrench icon (🔧).
3. From the **PowerPack Properties** page, click **Device Classes** in the Navbar on the left side of the page.
4. To add a device class, go to the **Available Device Classes** pane at the bottom of the page. Find the device class you want to include and click its lightning bolt icon (⚡). The content will be moved to the top pane and included in the PowerPack.

NOTE: If a device is no longer collecting, check to see if the device tags have been changed and no longer match the tags in the credential for selective discovery.

© 2003 - 2022, ScienceLogic, Inc.

All rights reserved.

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com

ScienceLogic

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010