SNMP Overview and Configuration

Simple Network Management Protocol (SNMP) allows network administrators to query devices for various information. Meraki allows SNMP polling to gather information either from the dashboard or directly from MR access points, MS switches, and MX security appliances. Third party network monitoring tools can use SNMP to monitor certain parameters. This document will outline Dashboard configuration and discuss the information that can be gathered.

Overview

The following information can be polled from the Dashboard:

- Device MAC address
- Device Serial number
- Device Name
- Device Status (Online or Offline)
- Device Last Contacted - Date and Time
- Mesh Status (Gateway or Repeater)
- Public IP Address
- Product Code (e.g. MR18-HW)
- Product Description (e.g. Meraki Cloud-controller 802.11n AP)
- Name of the Network that the device resides in (Dashboard Network)
- Packets/Bytes In/Out on each physical interface

Note: Exact information varies between the three product lines. Not all of the above information can be polled from devices locally.

Standard MIBs

Meraki devices support the majority of OIDs located within the subset of the following MIBs:

- SNMPv2-MIB .1.3.6.1.2.1.1
- IF-MIB .1.3.6.1.2.1

These MIBs are not proprietary and therefore are available on most network monitoring systems.

Proprietary MIB

Additional Meraki specific information can be found in the MERAKI-CLOUD-CONTROLLER-MIB, which is located under Organization > Settings > SNMP. Please note that this MIB is used to poll Dashboard, not Meraki devices locally. Please reference the Standard MIBs above for options when polling local devices.
Configuration

The Dashboard can be configured for SNMP polling from Organization > Settings > SNMP. Once SNMP has been enabled you will be able to send the SNMP requests to the host that is defined directly under the enable setting. It also defines the community string and provides a sample command to extract information via SNMP requests.

Below is a screenshot of the Dashboard configuration of SNMP:

![SNMP Configuration Screenshot]

**Note:** The hostname used for all SNMP requests should be `snmp.meraki.com`. Please avoid using organization-specific hostnames that use the format `nXXX.meraki.com` like `n123.meraki.com`.

IP restrictions can be configured to restrict SNMP access to particular sources. SNMP versions 1 and 2 send the community string in clear text, so IP restrictions would be useful to prevent unauthorized SNMP access if the community string is intercepted or learned by another party.

**Note:** Meraki SNMP access only supports get requests, so no changes to the Dashboard configuration can be made using SNMP. Meraki uses the read-only community string type. The following are supported when polling the dashboard: GetRequest and GetNextRequest snmp GET methods.

Individual Meraki devices can also be polled locally. In this scenario the SNMP traffic would stay within the local network and each device would need to be polled from the network management system. These settings can be found under Network-wide > General > SNMP:

![SNMP Access Configuration Screenshot]
Using SNMP to directly poll individual devices provides the ability to choose between SNMP v1/v2c or v3. Version 3 adds the ability to encrypt the communication between the manager (network management system) and the agent (Meraki device). SNMP v1/v2c send the community string in plain text. If v3 is selected, you will need to configure a username and password. When using v3, Cisco Meraki devices will use SHA1 for authentication and DES for privacy, with the configured password used for both.

**Note:** You will not be able to poll repeaters using SNMP as they do not have an IP address on the Local LAN

### Polling from Outside the Network

By default, Meraki devices cannot be polled via SNMP from outside the local IP network. However, after setting up the community string, an MX can be configured to allow polling from a remote IP whitelist, configured under **Organization > Configure > Settings > SNMP**.

### Testing SNMP

Multiple tools exist to test SNMP functionality on various operating systems. Windows based operating systems can use an open source tool called net-snmp. Follow these instructions to install net-snmp and test SNMP access to the Cloud Controller:

1. [Download](#) net-snmp and install
2. Save the MIB file from the Dashboard (**Organization > Settings > SNMP > Download MIB**) to the proper net-snmp directory: `C:\usr\share\snmp\mibs\` by default
3. Copy and paste the sample `snmp walk` from your Controller (**Organization > Settings > SNMP**) and place it in a command prompt

On a Linux-based system you can install a SNMP package to achieve the same results. Follow these instructions on Debian based systems:

1. Install snmpd. For systems with `apt-get`:
   ```
   sudo apt-get install snmpd
   ```
2. Save the MIB file from the dashboard (**Organization > Settings > SNMP > Download MIB**) to the proper `snmpd` directory
3. Copy and paste the example `snmpwalk` command from your Controller (**Organization > Settings > SNMP**) and place it in a terminal

Testing SNMP to individual MR’s is exactly the same, but you will use `snmpwalk` to each individual MR. For example if you have two MR access points you would need to use `snmpwalk` twice (MR 1 IP: 192.168.100.15, MR2 IP: 192.168.100.16):

```
snmpwalk -v2c -t 10 -c meraki 192.168.100.15:161
snmpwalk -v2c -t 10 -c meraki 192.168.100.16:161
```

Keep in mind that the `snmpwalk` command would need to modified based upon the version of SNMP that was configured. In the above example, SNMP v2c is being used, with a community string of ‘meraki’, and the port has been left at a default of 161.

The command below is an example `snmpwalk` for SNMPv3, using the username and passphrase configured in **Network-wide > General**:
```bash
snmpwalk -v3 -t 10 -a sha -A <passphrase> -u <username> -x des -X <passphrase> -l authpriv <IP address>
```

### SNMP Timeouts

In some circumstances it can take several seconds for our SNMP agent to initially query and build a database of devices for an organization. To give the agent enough time to respond, in the example commands above we have set a timeout of 10 seconds. If you are using proprietary network monitoring software such as WhatsUp Gold, the SNMP timeout should also be configured as the defaults are often too low.

### Sample Output

The following is sample output from the `snmpwalk` command run locally, to a local MR 16:

```bash
sysadmin@ubuntu:~$ snmpwalk -v2c -c meraki 192.168.100.15:161
iso.3.6.1.2.1.1.1.0 = STRING: "Meraki MR16 Cloud Managed AP"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.29671.2.13
iso.3.6.1.2.1.1.15.0 = STRING: "MR16"
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.2.1.31
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.2.840.10036
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The MIB module to describe generic objects for network interface sub-layers"
iso.3.6.1.2.1.1.9.1.4.4 = STRING: "The MIB module for managing 802.11 implementations"
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.1.0 = INTEGER: 3
iso.3.6.1.2.1.1.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.1.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.1.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.1.1.1.2.1 = STRING: "wired0"
iso.3.6.1.2.1.1.1.1.2.2 = STRING: "wifi0"
iso.3.6.1.2.1.1.1.1.2.3 = STRING: "wifi1"
iso.3.6.1.2.1.1.1.1.3.1 = INTEGER: 6
iso.3.6.1.2.1.1.1.1.3.2 = INTEGER: 71
iso.3.6.1.2.1.1.1.1.3.3 = INTEGER: 71
iso.3.6.1.2.1.1.1.1.4.1 = INTEGER: 1
iso.3.6.1.2.1.1.1.1.4.2 = INTEGER: 1
iso.3.6.1.2.1.1.1.1.4.3 = INTEGER: 1
iso.3.6.1.2.1.1.1.1.5.1 = Gauge32: 0
iso.3.6.1.2.1.1.1.1.5.2 = Gauge32: 0
iso.3.6.1.2.1.1.1.1.5.3 = Gauge32: 0
iso.3.6.1.2.1.1.1.1.6.1 = Hex-STRING: 00 18 0A 34 36 C4
iso.3.6.1.2.1.1.1.1.6.2 = Hex-STRING: 00 18 0A 34 36 C4
iso.3.6.1.2.1.1.1.1.6.3 = Hex-STRING: 02 18 1A 34 36 C4
iso.3.6.1.2.1.1.1.1.7.1 = INTEGER: 1
iso.3.6.1.2.1.1.1.1.7.2 = INTEGER: 1
iso.3.6.1.2.1.1.1.1.7.3 = INTEGER: 1
iso.3.6.1.2.1.1.1.1.8.1 = INTEGER: 1
iso.3.6.1.2.1.1.1.1.8.2 = INTEGER: 1
```
iso.3.6.1.2.1.2.1.8.3 = INTEGER: 1
iso.3.6.1.2.1.2.1.9.1 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.2.1.9.2 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.2.1.9.3 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.2.1.10.1 = Counter32: 77152729
iso.3.6.1.2.1.2.1.10.2 = Counter32: 70138554
iso.3.6.1.2.1.2.1.10.3 = Counter32: 17035160
iso.3.6.1.2.1.2.1.11.1 = Counter32: 0
iso.3.6.1.2.1.2.1.11.2 = Counter32: 0
iso.3.6.1.2.1.2.1.11.3 = Counter32: 0
iso.3.6.1.2.1.2.1.12.1 = Counter32: 0
iso.3.6.1.2.1.2.1.12.2 = Counter32: 0
iso.3.6.1.2.1.2.1.12.3 = Counter32: 0
iso.3.6.1.2.1.2.1.13.1 = Counter32: 0
iso.3.6.1.2.1.2.1.13.2 = Counter32: 0
iso.3.6.1.2.1.2.1.13.3 = Counter32: 0
iso.3.6.1.2.1.2.1.14.1 = Counter32: 0
iso.3.6.1.2.1.2.1.14.2 = Counter32: 0
iso.3.6.1.2.1.2.1.14.3 = Counter32: 0
iso.3.6.1.2.1.2.1.16.1 = Counter32: 16102848
iso.3.6.1.2.1.2.1.16.2 = Counter32: 35646246
iso.3.6.1.2.1.2.1.16.3 = Counter32: 1540911
iso.3.6.1.2.1.2.1.17.1 = Counter32: 0
iso.3.6.1.2.1.2.1.17.2 = Counter32: 97878
iso.3.6.1.2.1.2.1.17.3 = Counter32: 2728
iso.3.6.1.2.1.2.1.18.1 = Counter32: 0
iso.3.6.1.2.1.2.1.18.2 = Counter32: 0
iso.3.6.1.2.1.2.1.18.3 = Counter32: 0
iso.3.6.1.2.1.2.1.19.1 = Counter32: 0
iso.3.6.1.2.1.2.1.19.2 = Counter32: 0
iso.3.6.1.2.1.2.1.19.3 = Counter32: 0
iso.3.6.1.2.1.2.1.20.1 = Counter32: 0
iso.3.6.1.2.1.2.1.20.2 = Counter32: 0
iso.3.6.1.2.1.2.1.20.3 = Counter32: 0
iso.3.6.1.2.1.2.1.21.1 = Gauge32: 0
iso.3.6.1.2.1.2.1.21.2 = Gauge32: 0
iso.3.6.1.2.1.2.1.21.3 = Gauge32: 0
iso.3.6.1.2.1.2.1.22.1 = OID: ccitt.0
iso.3.6.1.2.1.2.1.22.2 = OID: ccitt.0
iso.3.6.1.2.1.2.1.22.3 = OID: ccitt.0
iso.3.6.1.2.1.31.1.1.1.1.1.1 = STRING: "wired0"
iso.3.6.1.2.1.31.1.1.1.1.1.2 = STRING: "wifi0"
iso.3.6.1.2.1.31.1.1.1.1.1.3 = STRING: "wifi1"
iso.3.6.1.2.1.31.1.1.1.1.2.1 = Counter32: 0
iso.3.6.1.2.1.31.1.1.1.1.2.2 = Counter32: 0
iso.3.6.1.2.1.31.1.1.1.1.2.3 = Counter32: 0
iso.3.6.1.2.1.31.1.1.1.1.3.1 = Counter32: 0
iso.3.6.1.2.1.31.1.1.1.1.3.2 = Counter32: 0
iso.3.6.1.2.1.31.1.1.1.1.3.3 = Counter32: 0
iso.3.6.1.2.1.31.1.1.1.1.4.1 = Counter32: 0
iso.3.6.1.2.1.31.1.1.1.1.4.2 = Counter32: 0
iso.3.6.1.2.1.31.1.1.1.1.4.3 = Counter32: 0
iso.3.6.1.2.1.31.1.1.1.1.5.1 = Counter32: 0
iso.3.6.1.2.1.31.1.1.1.1.5.2 = Counter32: 0
After verifying that SNMP is functioning properly, the network management system (NMS) can be configured to poll either the dashboard, MR's, or both. Consult your NMS documentation for configuration steps to add a new managed device (MR/dashboard).

### SNMP Traps

SNMP traps allow real-time alerting of specific network scenarios. SNMP traps will always be forwarded from the Meraki Cloud. A complete list of traps is available in the Meraki MIB. SNMP traps use SHA1 for authentication and AES for privacy.

**Note:** The order of variables within an SNMP trap may be different than expected. Administrators should look at the trap message themselves before configuring any parsers that rely on a particular variable order.

### Enabling Traps

Contact Meraki Support to have the feature enabled on the desired organization. Navigate to Network-wide > Alerts. Under SNMP traps, fill out the required fields. Traps will be sent from the Meraki Cloud. If the receiving server is behind a NAT, port forwards will have to be used. The test trap button can be used to verify that traps are being sent and received successfully.
Note that the Receiving server IP must be a public IP.

Defining traps to be sent

Once traps have been enabled, alerts that will trigger a trap to be sent are configured under the Alerts Settings section found under Network-wide > Alerts. To configure an SNMP trap to be sent, click on Show additional recipients for a selected alert and add "snmp" as a recipient. Alternatively, you can add "snmp" as a Default recipient. Once configured, both an email and a trap will be sent when an event is detected.

Note: Since traps are sent from Dashboard, any time sensitive traps will begin counting from the moment that Dashboard detects the event.