

The background is a vibrant, abstract graphic. It features a central bright white light source from which numerous colorful rays emanate, creating a sunburst or starburst effect. The rays transition through a spectrum of colors including yellow, orange, red, and various shades of blue and green. Overlaid on this are several large, semi-transparent, wavy shapes in similar color tones, giving the overall image a sense of motion and energy.

cisco *Live!*

Let's go

#CiscoLive



The bridge to possible

# Enhanced Automation of a Meraki-based Environment

Jason Davis / Distinguished Engineer  
@SNMPguy  
DEVNET-2200



# Cisco Webex App

## Questions?

Use Cisco Webex App to chat with the speaker after the session

## How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 9, 2023.



<https://ciscolive.ciscoevents.com/ciscolivebot/#DEVENT-2200>

# Agenda

- Situation
- Proposed Solution
- Imagineering
- Refining
- Where the code is

# Situation

## A real-life scenario

- You are new to Meraki gear and happy with it
- Look -- Here come examiners to do audits
- Time to pull out proof of compliance!

“Oh, that’s easy – let’s just download the configs”



Bruh, Meraki equipment is cloud-managed.

There are *NO* configs to download and review.

This isn't IOS-XE or NX-OS!



“Oh, that’s OK, we’ll just give the Auditor login access to our thousands of Meraki networks and devices

They can check themselves...”



$$\lim_{x \rightarrow \infty} f(x_{score}) = \frac{1}{x}$$

Math of Auditors



# Imagineering

## Getting the Requirements

- We must produce an approximation or analog of a device config
- Data format needs to be consumable and programmatically viable
- We need to store the data securely as it contains sensitive device configuration info
- Diffs must be able to be generated between runs
- We need to define how often to collect and analyze



# How Researched

- Meraki Dashboard API documentation  
<https://developer.cisco.com/meraki/api-v1/>



A screenshot of the Meraki Dashboard API documentation page on the Cisco Developer Hub. The page has a dark header with "DEVNET" and navigation links like "SIGN UP FREE" and "LOG IN". Below the header, the breadcrumb "Meraki &gt; API &gt; Dashboard API" is shown. The left sidebar contains a search bar and a table of contents with sections like "Meraki Dashboard API", "Introduction", "Authentication", "Getting Started", "API Changelog", "Guides", "API Reference", and "Overview". The main content area is titled "Meraki Dashboard API" and describes it as a RESTful API for managing Meraki networks. It includes a green icon with a code symbol, a section "What can you do with it?" with a bulleted list of capabilities, and a section "What's New in v1" with details about new features and improvements.

# How Researched

- Postman collection  
<http://cs.co/900438EmA>

A screenshot of the Postman web application interface. The top navigation bar includes links for Product, Pricing, Enterprise, Resources and Support, and Explore. A search bar and 'Sign In'/'Sign Up for Free' buttons are on the right. The left sidebar shows a 'Collections' list with 'Meraki Dashboard API - v1.30.0' selected. The main panel displays the 'Meraki Dashboard API - v1.30.0' collection details, including an 'Authorization' tab with a warning about sensitive data and a table for API key configuration. The right sidebar shows 'Documentation' and 'NEXT IN THIS COLLECTION' links.

Product Pricing Enterprise Resources and Support Explore

Search Postman

Sign In Sign Up for Free

Cisco Meraki's Public Workspace

Meraki Dashboard API - v1.30.0

Share Fork 2.06k Watch 713 Run

Authorization Pre-request Script Tests Variables

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

Type API Key

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#).

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#).

Key	X-Cisco-Meraki-API-Key
Value	{{apiKey}}
Add to	Header

Documentation

The Cisco Meraki Dashboard API is a modern REST API based on the OpenAPI specification.

Date: 04 January, 2023

[Recent Updates](#)

[API Documentation](#)

[Community Support](#)

[Meraki Homepage](#)

Contact Support:  
Name: Meraki Developer Community

Authorization API Key

Key	X-Cisco-Meraki-API-Key
Value	{{apiKey}}

NEXT IN THIS COLLECTION

- PRODUCTS
- GENERAL
- GET List the Organizations
- GET List the Networks in an Organization
- GET List the Devices in an Organization
- GET List the Clients in a Network

# How Researched

- OpenAPI spec published to GitHub  
<https://github.com/meraki/openapi>



```
{
  "swagger": "2.0",
  "info": {
    "version": "1.29.0",
    "title": "Meraki Dashboard API",
    "description": "The Cisco Meraki Dashboard API is a modern REST API based on the OpenAPI specification.\n\n Date: 04 January, 2023\n\n [Recent Updates](https://meraki.io/whats-new/)\n\n---\n\n[API Documentation](https://meraki.io/api/)\n\n[Community Support](https://meraki.io/community)\n\n[Meraki Homepage](https://www.meraki.com)\n",
    "contact": {
      "name": "Meraki Developer Community",
      "url": "https://meraki.io/community"
    }
  },
  "host": "api.meraki.com",
  "basePath": "/api/v1",
  "schemes": [
    "https"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "securityDefinitions": {
    "meraki_api_key": {
      "type": "apiKey",
      "name": "X-Cisco-Meraki-API-Key",
      "in": "header"
    }
  },
  "security": [
    {
      "meraki_api_key": [
        ]
      }
    ]
  },
  "paths": {
    "/administered/identities/me": {
      "get": {
        "description": "Returns the identity of the current user.",
        "operationId": "getAdministeredIdentitiesMe",
        "responses": {
          "200": {
            "description": "Successful operation",
            "schema": {
              "type": "object",
              "properties": {
                "name": {
                  "type": "string",
                  "description": "Username"
                },
                "email": {
                  "type": "string",
                  "description": "Email address"
                }
              }
            }
          }
        }
      }
    }
  }
}
```

# Parameterized API URLs

- Recognize data is indexed in several forms:  
organization, network, device

[, Interface, clientId, portId, vlanId, etc]

GET networks

<https://api.meraki.com/api/v1/organizations/:organizationId/networks>

GET devices

<https://api.meraki.com/api/v1/organizations/:organizationId/devices>

# First Implementation

- Wrote a Python program to collect all network and device instances in an org; sequentially polled each API endpoint from OpenAPI spec

Logic	operationId	tags	description
	getOrganizationNetworks	['organizations', 'configure', 'networks']	List the networks that the user has priv
	getOrganizationConfigTemplates	['organizations', 'configure', 'configTemplates']	List the configuration templates for thi
	getOrganizationLicensesOverview	['organizations', 'monitor', 'licenses', 'overview']	Return an overview of the license state
	getOrganizationInventoryDevices	['organizations', 'configure', 'inventoryDevices']	Return the device inventory for an orga
	getOrganizationDevices	['organizations', 'configure', 'devices']	List the devices in an organization
	getOrganizationAdmins	['organizations', 'configure', 'admins']	List the dashboard administrators in th
	getOrganizationBrandingPolicies	['organizations', 'configure', 'brandingPolicies']	List the branding policies of an organiz
	getOrganizationBrandingPoliciesPriorities	['organizations', 'configure', 'brandingPolicies', 'priorities']	Return the branding policy IDs of an or
	getOrganizationSaml	['organizations', 'configure', 'saml']	Returns the SAML SSO enabled setting
	getOrganizationSamlIdps	['organizations', 'configure', 'saml', 'idps']	List the SAML IdPs in your organization
	getOrganizationSamlRoles	['organizations', 'configure', 'samlRoles']	List the SAML roles for this organizatio
	getOrganizationSnmp	['organizations', 'configure', 'snmp']	Return the SNMP settings for an organ
	getOrganizationApplianceSecurityIntrusion	['appliance', 'configure', 'security', 'intrusion']	Returns all supported intrusion setting:
	getOrganizationApplianceVpnThirdPartyVPNPeers	['appliance', 'configure', 'vpn', 'thirdPartyVPNPeers']	Return the third party VPN peers for ar

# Results - First Implementation



Archived



Diff-able



Scannable

- Extract Meraki device **settings** from the Meraki Dashboard API (REST) and store as JSON records in text files of a local, secure git repo
- Git provides change monitoring; JSON Path Queries enable compliance checking against the settings
- Reports are generated for review
- Light-weight Linux VM on-prem integrates to Meraki Dashboard API as SaaS-friendly deployment

```
[jadavis@cdtao-centos L_6991]$ cat network-L_6991
{
  "id": "L_6991",
  "organizationId": "480",
  "type": "combined",
  "productTypes": [
    "appliance",
    "switch"
  ],
  "url": "https://meraki.com/MN_6991/page/manage/usage/list",
  "configTemplateId": "L_6991",
  "name": "MN_Home M",
  "timeZone": "US/Eastern",
  "tags": "BTES CMS_Managed CRITICAL_SITE",
  "disableMyMerakiCom": true,
  "disableRemoteStatusPage": true,
  "enrollmentString": null
}
```

# Results - First Implementation

With the thousands of networks and devices, coupled with hundred+ API endpoints to parse...

...the first experience due to Meraki API Rate Limiting (5/sec\*)

57 hours

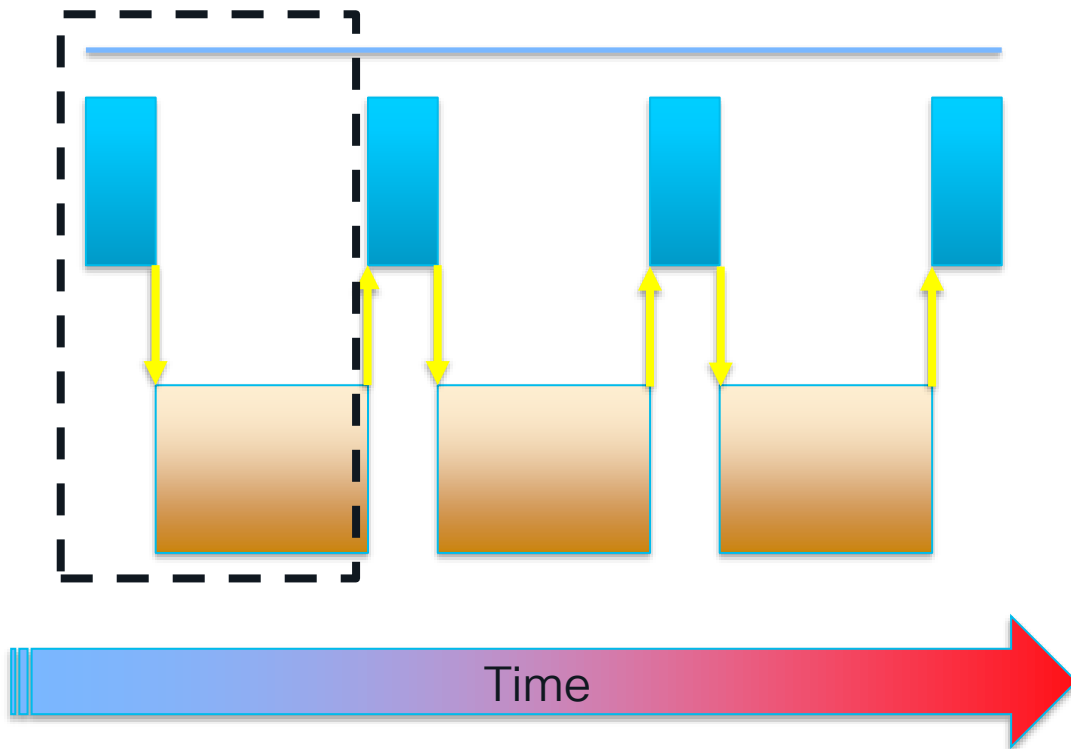




# Why So? Original view of sequential work

  
Processing

  
I/O  
Network, Disk



# Example - Traditional Sequential Processing

getcard-requests.py > ...

```
1 import requests
2 import time
3
4 start_time = time.time()
5
6 for number in range(1, 105):
7     url = f'https://deckofcardsapi.com/api/deck/new/draw/'
8     resp = requests.get(url)
9     card = resp.json()
10    print(f'{card["cards"][0]["value"]} of {card["cards"][0]["suit"]}')
11
12 print(f'--- %s seconds ---' % (time.time() - start_time))
```

OLE

TERMINAL

```
2 of SPADES
7 of HEARTS
KING of SPADES
3 of CLUBS
8 of SPADES
4 of SPADES
3 of DIAMONDS
2 of DIAMONDS
6 of CLUBS
4 of HEARTS
3 of CLUBS
4 of CLUBS
3 of CLUBS
3 of CLUBS
8 of HEARTS
```

--- 12.645462989807129 seconds ---

12.64 seconds

# Imagineering - Try 2

- How to speed up?  
Ideally get this down to around 12 hours so running once a day is feasible
- Multiprocessing?  
Multithreading?  
Multiple pollers?

# Imagineering - Try 2

First, we need to recognize where the bottlenecks

- My program?

There are definite opportunities to optimize, but that's not the main culprit

- My environment?

No - throwing more compute, memory and faster network doesn't alleviate the cloud-based API server dependency

- The cloud API server dependency?

BINGO! I had 5 (now 10) API calls/sec allotted. Anything I optimize must work within that restriction

# Imagineering - Try 2

So, increasing the complexity with Multiprocessing, Multithreading or multiple pollers won't be effective

We need a solution to queue all the work and operate within the constraints of the API server rate-limiting

Leading us to...



# AsyncIO


- A type of concurrency that enables a single processor to handle mass amount of work – essentially cooperative multitasking
- Good for I/O-bound tasks
- As solution developers **we** define the queue of work and when it is OK to swap away to do other work
  - waiting for IO (network, disk)
  - API processing


<https://docs.python.org/3/library/asyncio.html>

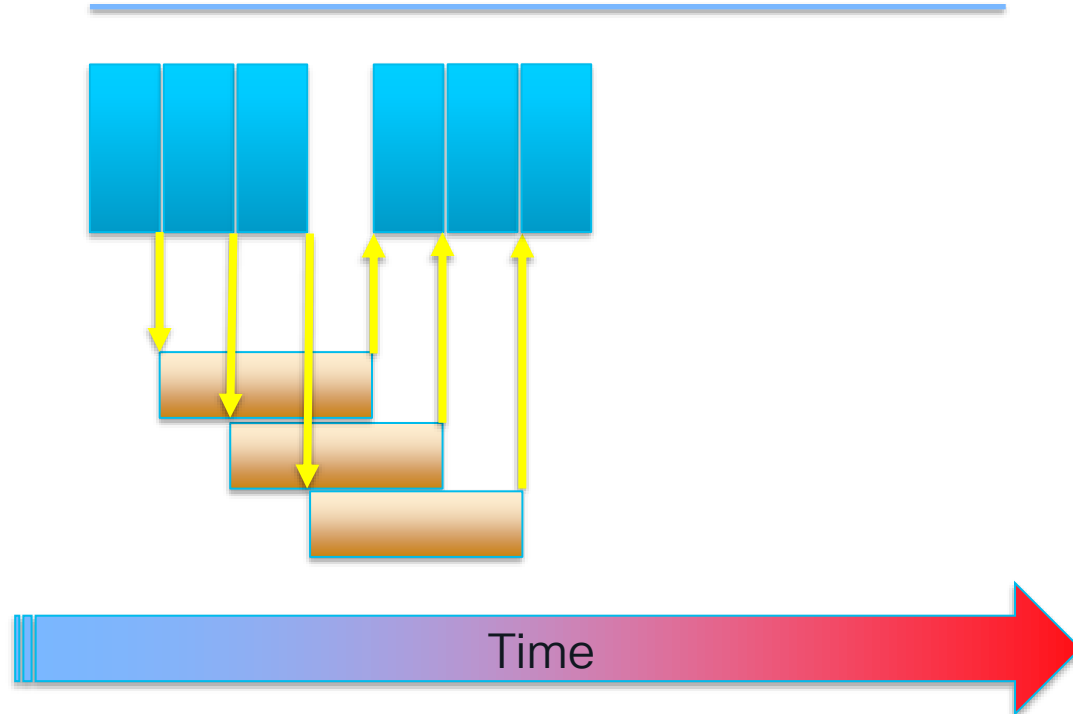
# AsyncIO

- You load up a 'queue' of work and the Python asyncio library uses an event loop to control processing and state of active/awaiting tasks
- You identify tasks that may take time to process, allowing asyncio opportunity to 'switch away' to the next task in the event loop queue

# With Cooperative Multitasking / asyncio

  
Processing

  
I/O  
Network, Disk





# Example - Cooperative Multitasking with asyncio

getcard-asyncio.py > ...

```
1 import aiohttp
2 import asyncio
3 import time
4
5 start_time = time.time()
6
7 async def main():
8     async with aiohttp.ClientSession() as session:
9         for number in range(1, 105):
10             url = f'https://deckofcardsapi.com/api/deck/new/draw/'
11             async with session.get(url) as resp:
12                 card = await resp.json()
13                 print(f'{card["cards"][0]["value"]} of {card["cards"][0]["suit"]}')
14
15 asyncio.run(main())
16
17 print(f'--- %s seconds ---' % (time.time() - start_time))
```

OLE

TERMINAL

5.99 seconds

52%  
improvement

```
QUEEN of HEARTS
ACE of HEARTS
10 of HEARTS
7 of CLUBS
8 of HEARTS
2 of HEARTS
4 of DIAMONDS
7 of DIAMONDS
4 of CLUBS
5 of HEARTS
```

--- 5.98675012588501 seconds ---

# Results after Optimization

- Using the same thousands of networks and devices and Meraki API Rate Limiting (5/sec\*), the second experience refactored with asyncio
- 13 hours

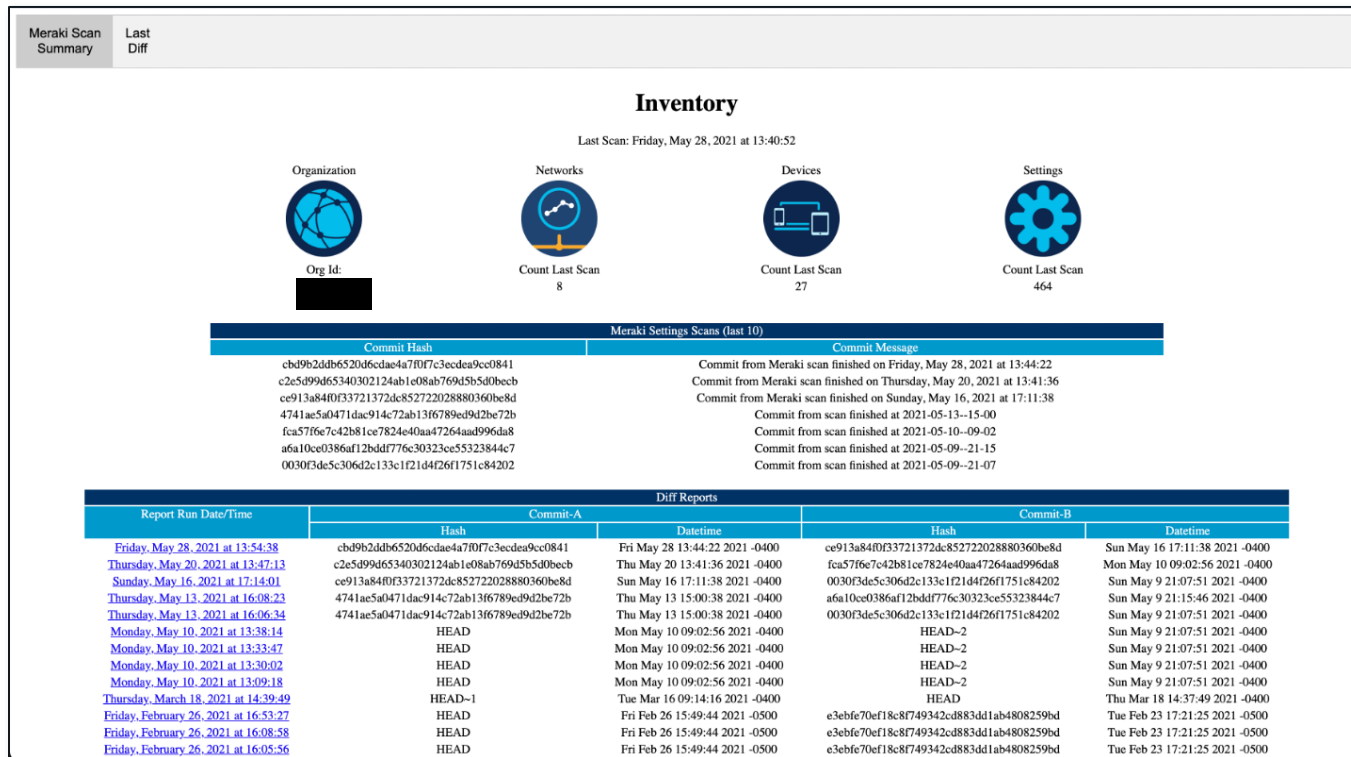


77% improvement

# Output of Meraki Settings Archive & Differ Project

Start Page

Scan Stats & List  
Diff Reports



# Output of Meraki Settings Archive & Differ Project

## Scan Results

Individual category  
(Org, Network, Device)  
settings changes  
identified

### Scan results

**The following settings/files were affected in last scan of  
commit HEAD at Fri May 28 13:44:22 2021 -0400 with  
commit HEAD~2 at Sun May 16 17:11:38 2021 -0400:**



[devices/Q2SW](#) [MX250/device\\_ApplianceDhcpSubnets](#)  
[networks/L\\_56](#) [8049 - Lyoli/network\\_FirmwareUpgrades](#)  
[networks/L\\_56](#) [8049 - Lyoli/network\\_FloorPlans](#)  
[networks/L\\_56](#) [8927 - Vegas Apartment/network\\_FirmwareUpgrades](#)  
[networks/L\\_56](#) [8854 - My network/network\\_Devices](#)  
[networks/L\\_56](#) [8854 - My network/network\\_SwitchStp](#)  
[networks/L\\_78](#) [8320 - DevNetLab/network\\_ApplianceFirewallInboundFirewallRules](#)  
[networks/L\\_78](#) [8514 - DevNetLab2/network\\_ApplianceFirewallOneToOneNatRules](#)  
[networks/L\\_78](#) [8514 - DevNetLab2/network\\_ApplianceSecurityIntrusion](#)  
[networks/L\\_78](#) [8514 - DevNetLab2/network\\_FirmwareUpgrades](#)  
[networks/L\\_78](#) [8515 - DevNetLab3/network\\_FirmwareUpgrades](#)  
[networks/N\\_56](#) [9127 - Nolan/network\\_FirmwareUpgrades](#)  
[org\\_Admins](#)  
[org\\_Devices](#)  
[org\\_DevicesStatuses](#)  
[org\\_InventoryDevices](#)

# Output of Meraki Settings Archive & Differ Project

## Diff Results

Settings Diffs  
identified and  
colorized

```
Diff of... networks/L_783[REDACTED]514 - DevNetLab2/network_FirmwareUpgrades.json

Commit A - HEAD scan datetime Fri May 28 13:44:22 2021 -0400
to
Commit B - HEAD~2 scan datetime Sun May 16 17:11:38 2021 -0400

Report Date - 20210528-135438

Files changed (1) show
networks/L_783[REDACTED]514 - DevNetLab2/network_FirmwareUpgrades.json CHANGED [Viewed]

1 {
2   "products": {
3     "appliance": {
4       "currentVersion": {
5         "id": 2009,
6         "firmware": "wired-14-53",
7         "releaseType": "legacy",
8         "shortName": "MX 14.53"
9       },
10      "lastUpgrade": {
11        "time": "",
12        "fromVersion": {},
13        "toVersion": {}
14      },
15      "nextUpgrade": {
16        "time": "",
17        "toVersion": {}
18      },
19      "availableVersions": [
20        {
21          "id": 2155,
22          "firmware": "wired-15-42-1",
23          "releaseType": "stable",
24          "shortName": "MX 15.42.1"
25        },
26        {
27 -       "id": 2170,
28 -       "firmware": "wired-16-2",
29 -       "releaseType": "beta",
30 -       "shortName": "MX 16.2"
31 -     },
32      ]
33    },
34    "cellularGateway": {
35      "currentVersion": {
36        "id": 1909,
37        "firmware": "Custom version (528ac28)"
38      }
39    }
40  }
41 }
```

# Meraki Settings Archive & Differ on Cisco DevNet Code Repo



<https://developer.cisco.com/network-automation/detail/bdbdb464-db3a-11eb-95a5-c6918c6fb71b/>

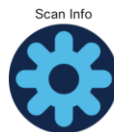
# Overall Org Compliance Assessment Scoring\*

## Compliance Assessment



ORGANIZATION:

48 [REDACTED]



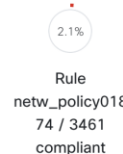
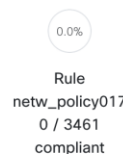
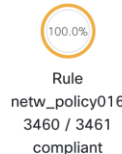
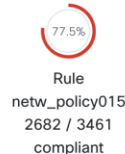
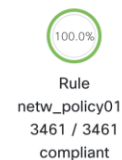
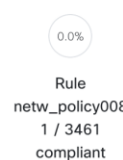
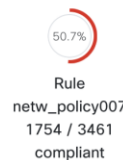
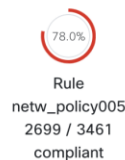
REPORT RUN

Tuesday, April 06, 2021 at 20:13:20  
Meraki Data Gathered:  
Sat Apr 3 22:15:31 2021 -0400

Detailed Compliance Info



[CLICK HERE](#)



Rule Number	Rule Description
Rule 1	Disable local and remote device status pages
Rule 2	Explicit cleanup rule added to end of every L3 FW policy
Rule 3	Explicit cleanup rule added to end of every L3 FW Group Policy
Rule 5	Intrusion detection and prevention is set to: Mode=Prevention, Ruleset=Security

# Individual Networks/Devices Compliance Assessment

	NetworkId	Location	Template	Rule 1	Rule 2	Rule 3	Rule 5	Rule 6	Rule 7	Rule 8	Rule 9	Rule 11	Rule 12	Rule 13	Rule 14	Rule 15	Rule 16	Rule 17	Rule 18
1	N_699	875 UN2	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
2	N_699	876 UN2	L_6	75	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
3	N_699	877 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
4	N_699	878 UN2	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
5	N_699	879 UN2	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
6	N_699	880 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
7	N_699	881 UN2	L_6	34	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
8	N_699	882 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
9	N_699	883 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
10	N_699	884 UN2	L_6	34	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
11	N_699	885 UN2	L_6	83	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
12	N_699	886 UN2	L_6	34	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
13	N_699	887 UN2	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
14	N_699	888 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
15	N_699	889 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
16	N_699	890 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
17	N_699	891 UN2	L_6	75	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
18	N_699	892 UN2	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
19	N_699	893 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
20	N_699	894 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
21	N_699	895 UN2	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
22	N_699	896 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
23	N_699	897 UN2	L_6	83	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
24	N_699	898 UN2	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
25	N_699	899 UN2	L_6	76	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
26	N_699	900 UN2	L_6	58	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
27	N_699	901 UNE	L_6	38	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
28	N_699	902 UN2	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
29	N_699	903 UN2	L_6	75	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
30	N_699	904 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
31	N_699	905 UN2	L_6	34	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
32	N_699	906 UN2	L_6	83	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
33	N_699	907 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
34	N_699	908 UN2	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
35	N_699	909 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
36	N_699	910 UNE	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
37	N_699	984 MN	L_6	30	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●





You can do IT too!

# Application/Challenge

- Watch for changes on the OpenAPI spec and API docs  
New features come out regularly - latest v1.30 January 4, 2023!
- Pay attention to the indexing of the API endpoints
- Remember you currently have a 10 calls/sec limit
- When dealing with large numbers of API endpoints and devices use asyncio to be more efficient  
Bonus - The Meraki Dashboard API Python library already uses asyncio internally - you only need wrappers for YOUR code

```
pip install meraki
```

# Continue your education



- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at [www.CiscoLive.com/on-demand](https://www.CiscoLive.com/on-demand)

# Fill out your session surveys!



Attendees who fill out a minimum of four session surveys and the overall event survey will get **Cisco Live-branded socks** (while supplies last)!



Attendees will also earn 100 points in the **Cisco Live Challenge** for every survey completed.



**These points** help you get on the leaderboard and increase your chances of winning daily and grand prizes

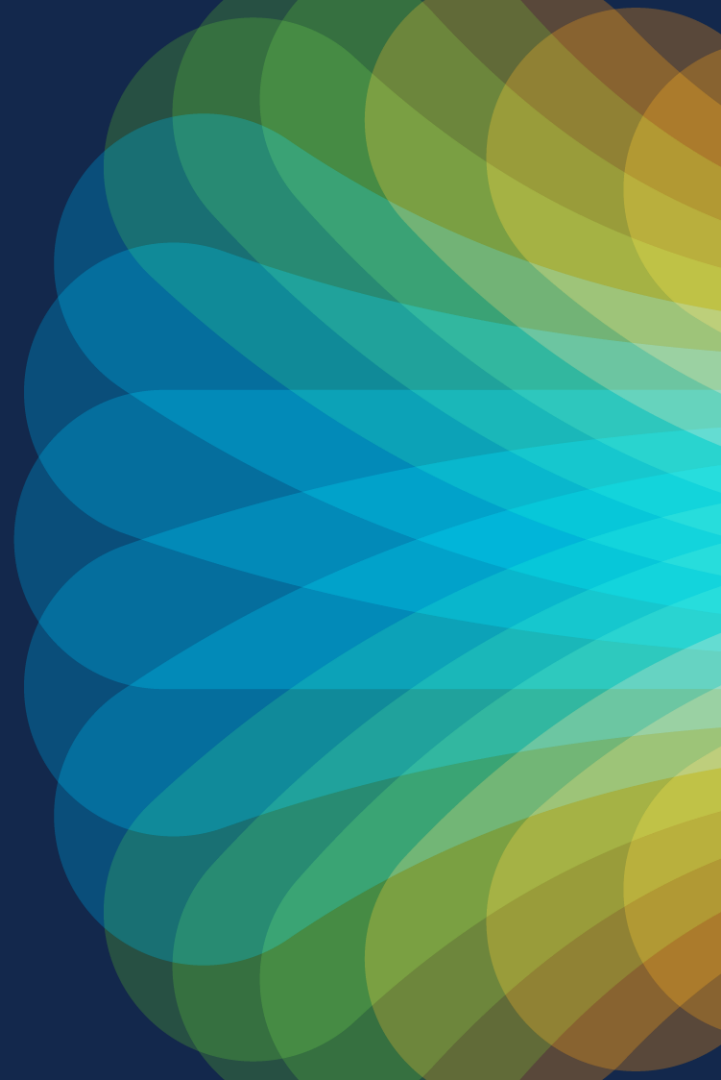


The bridge to possible

# Thank you

CISCO *Live!*

#CiscoLive



The background is a vibrant, abstract graphic. It features a central bright white light source from which numerous colorful rays emanate, creating a sunburst or starburst effect. The rays transition through a spectrum of colors including yellow, orange, red, and various shades of blue and green. Overlaid on this are large, flowing, wavy shapes in similar colors, giving the impression of liquid or smoke being illuminated by the light. The overall effect is energetic and celebratory.

cisco *Live!*

Let's go

#CiscoLive

# Cisco Live Challenge

Gamify your Cisco Live experience!  
Get points for attending this session!

## How:

- 1 Open the Cisco Events App.
- 2 Click on 'Cisco Live Challenge' in the side menu.
- 3 Click on View Your Badges at the top.
- 4 Click the + at the bottom of the screen and scan the QR code:

