
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 Live!

Cisco Webex App

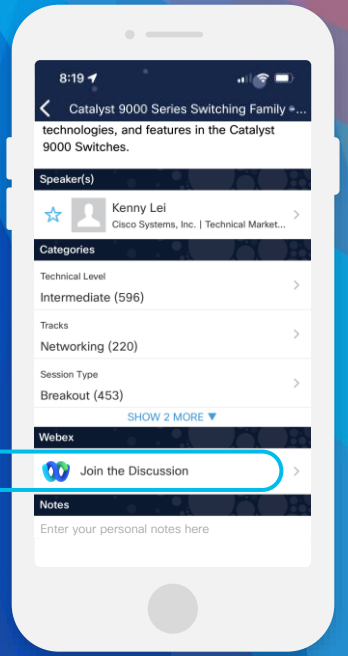
Questions?

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!

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…”

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

Math of Auditors

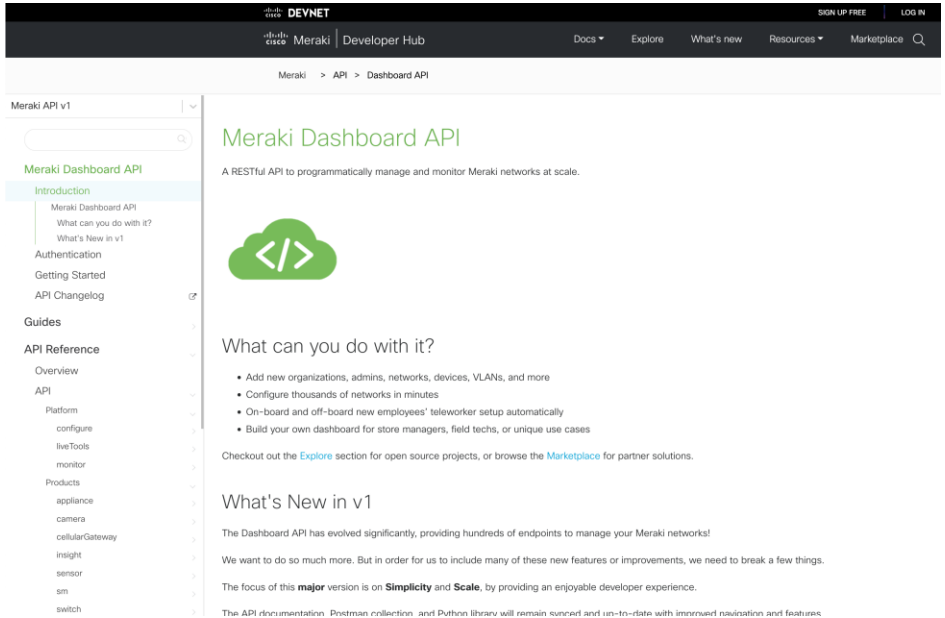
Imagineering

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

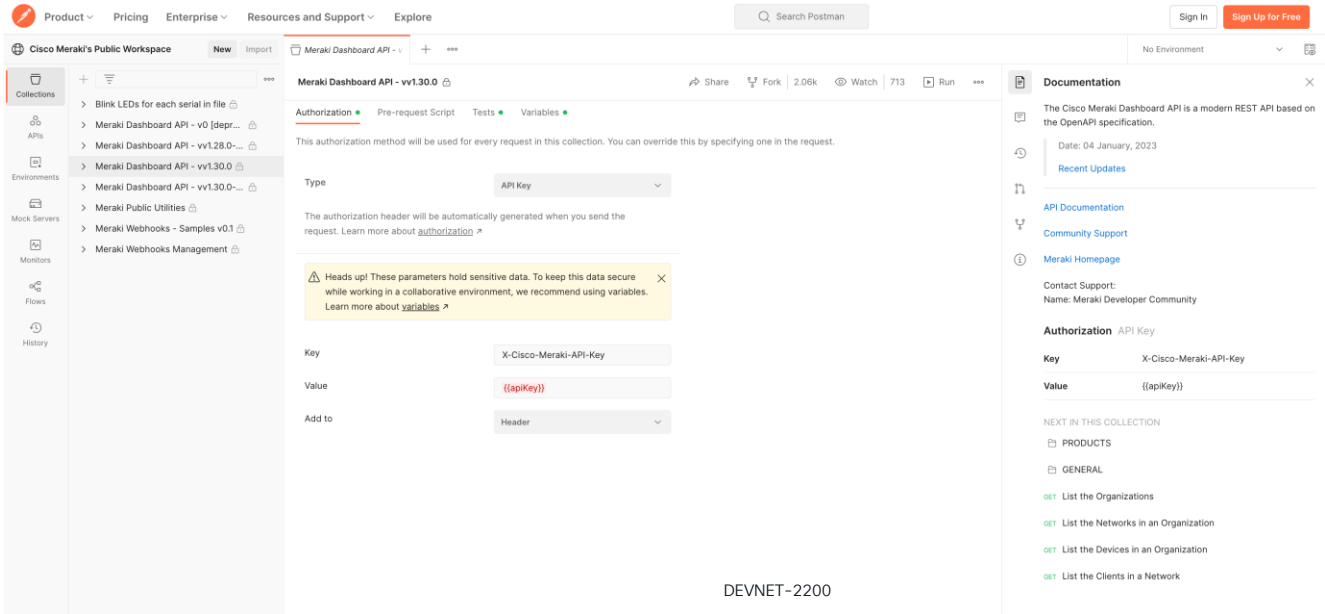
How Researched

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



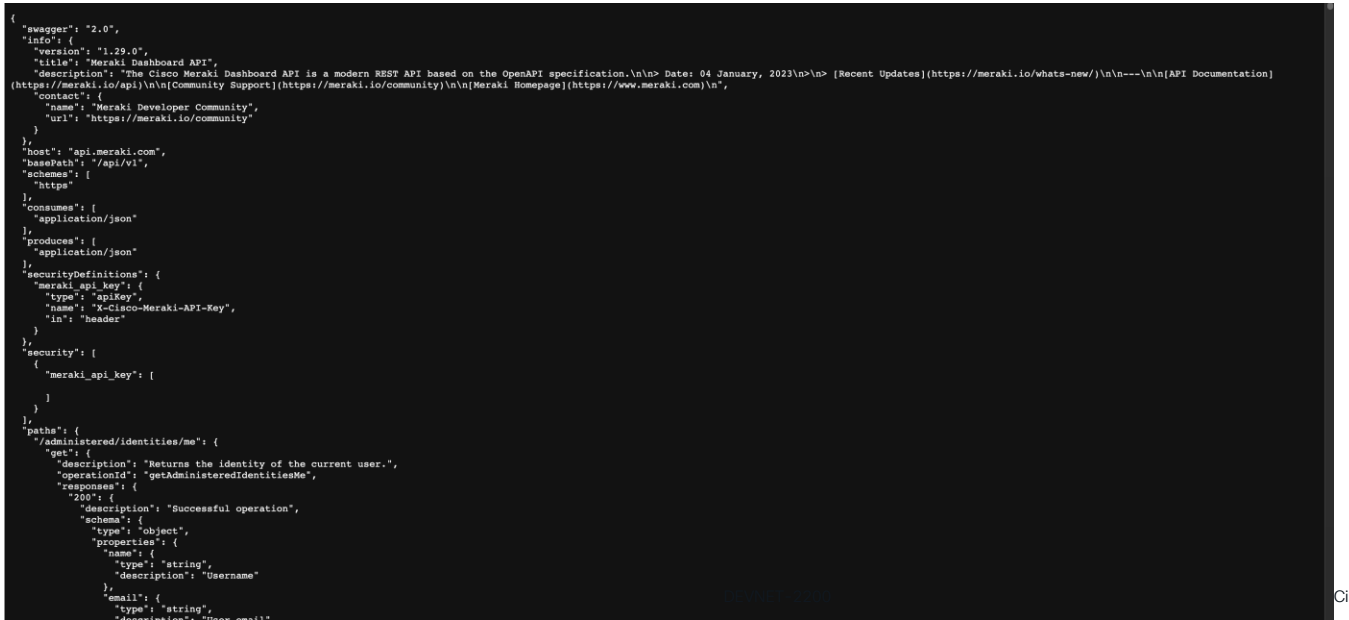
How Researched

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



How Researched

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



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 organizatior
	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

Results – First Implementation

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

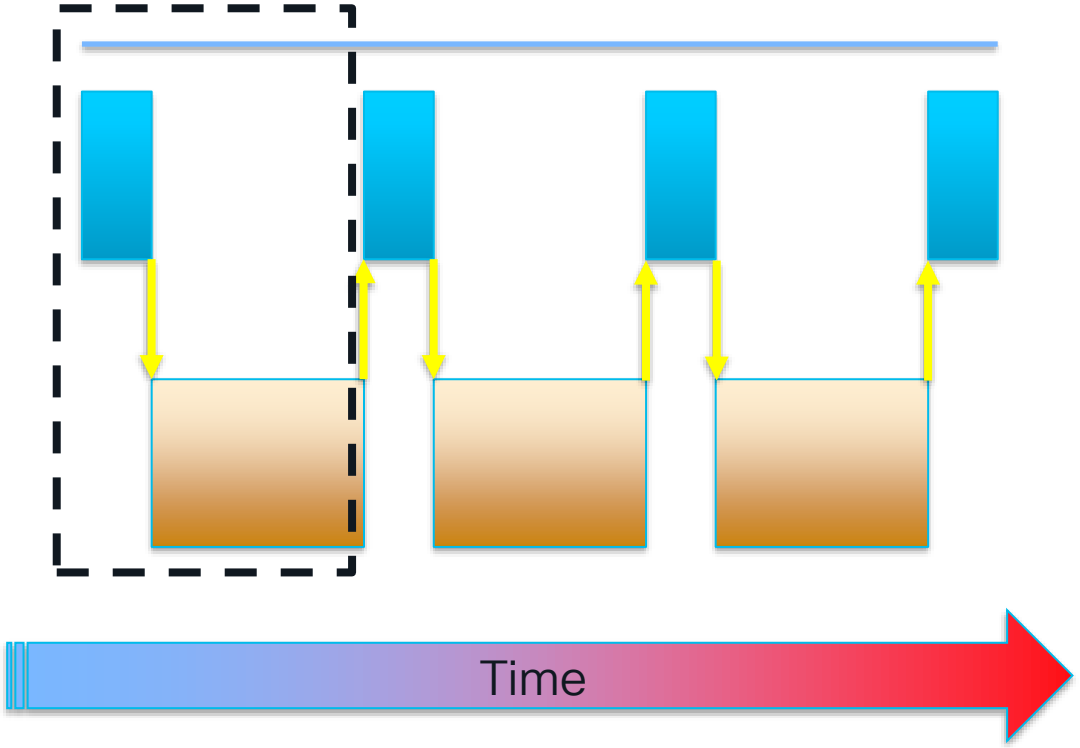
...the first experience due to Meraki API Rate Limiting (5/sec*)

57 hours

Why So? Original view of sequential work

Processing

I/O
Network, Disk



Example - Traditional Sequential Processing

getcard-requests.py > ...

```python
import requests
import time

start_time = time.time()

for number in range(1, 105):
    url = f'https://deckofcardsapi.com/api/deck/new/draw/'
    resp = requests.get(url)
    card = resp.json()
    print(f'{card["cards"][0]["value"]} of {card["cards"][0]["suit"]}')

print("--- %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

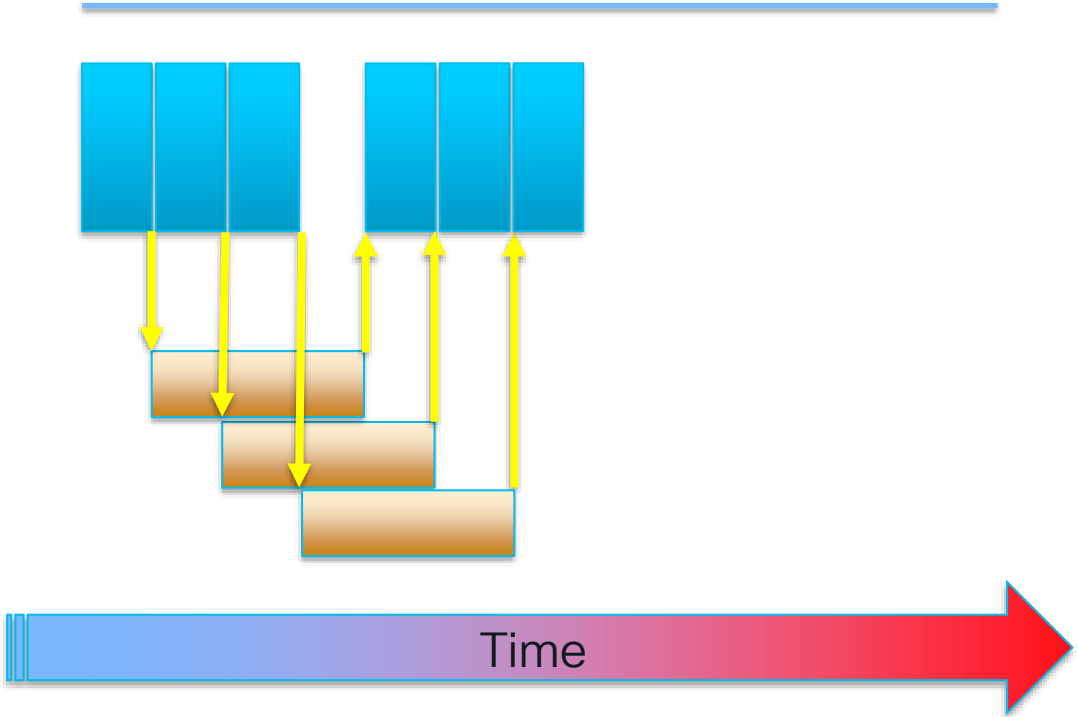
- You load up a ‘queue’ of work and the Python asyncio library uses an event loop to control processing and state of active/awaiting tasks
- You identify tasks that may take time to process, allowing asyncio opportunity to ‘switch away’ to the next task in the event loop queue

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

1

2

3

4 & 5

6

OLE TERMINAL

```
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 ---
```

5.99 seconds

52% improvement

Results after Optimization

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

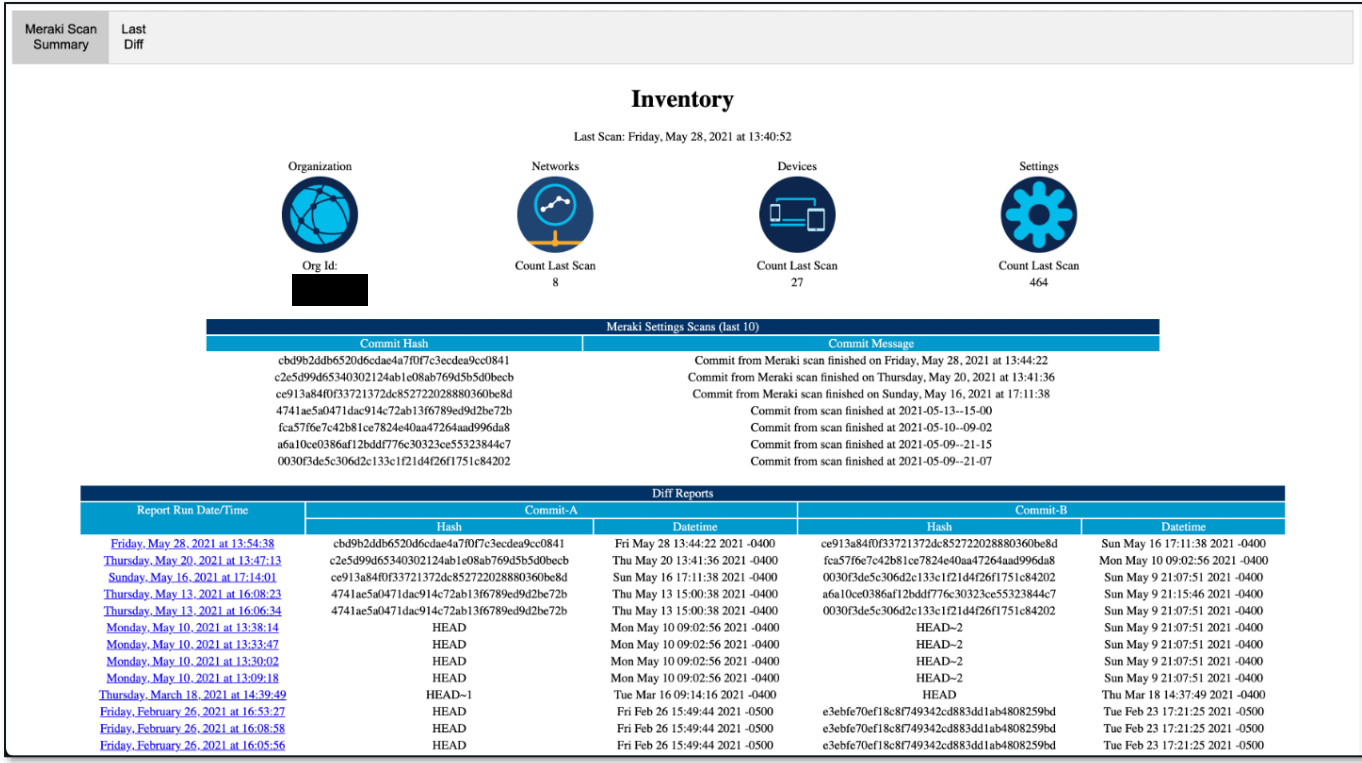
- 13 hours

77% improvement

Output of Meraki Settings Archive & Differ Project

Start Page

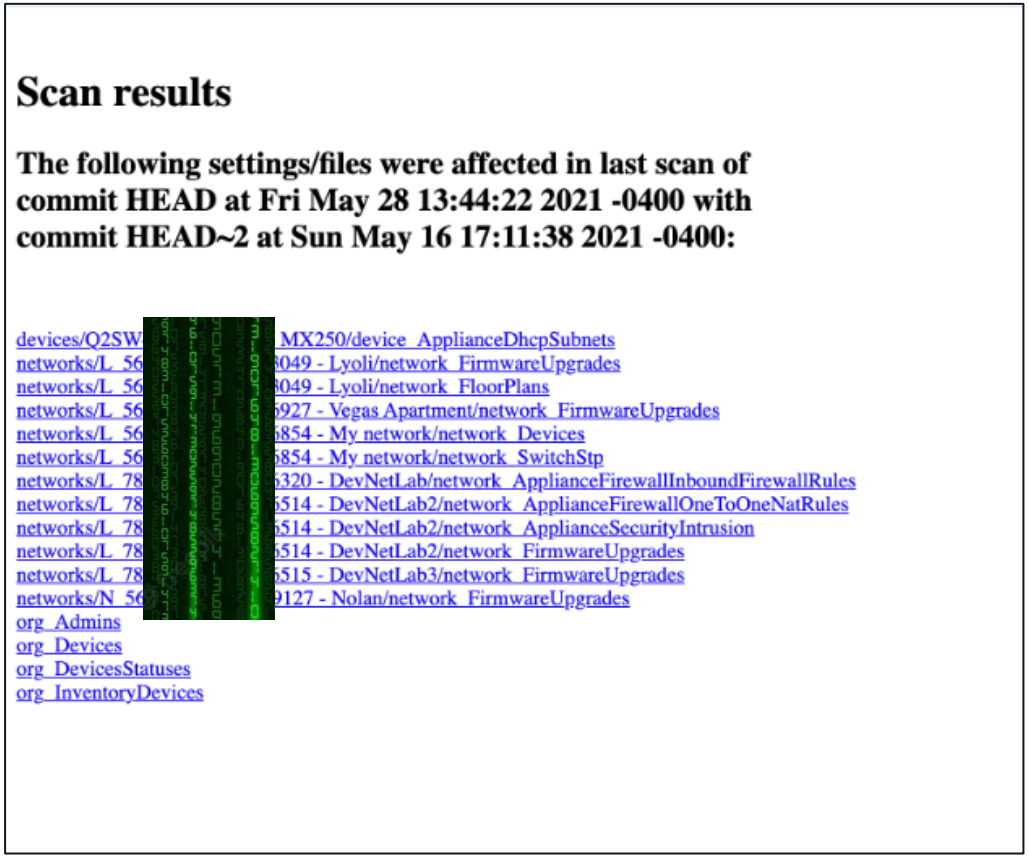
Scan Stats & List Diff Reports



Output of Meraki Settings Archive & Differ Project

Scan Results

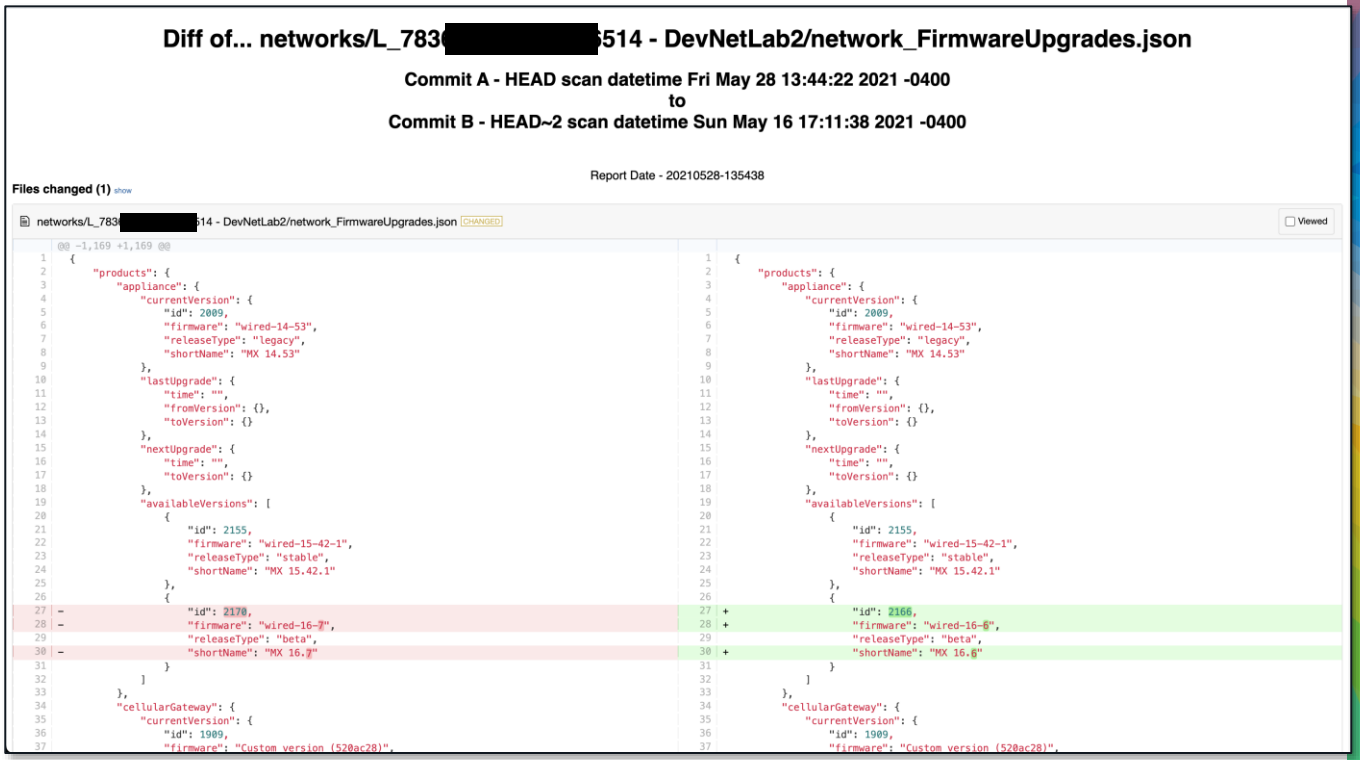
Individual category (Org, Network, Device) settings changes identified



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

```
@@ -1,169 +1,169 @@
 {
     "products": {
         "appliance": {
             "currentVersion": {
                 "id": 2009,
                 "firmware": "wired-14-53",
                 "releaseType": "legacy",
                 "shortName": "MX 14.53"
             },
             "lastUpgrade": {
                 "time": "",
                 "fromVersion": {},
                 "toVersion": {}
             },
             "nextUpgrade": {
                 "time": "",
                 "toVersion": {}
             },
             "availableVersions": [
                 {
                     "id": 2155,
                     "firmware": "wired-15-42-1",
                     "releaseType": "stable",
                     "shortName": "MX 15.42.1"
                 },
                 {
-                    "id": 2170,
-                    "firmware": "wired-16-7",
+                    "id": 2166,
+                    "firmware": "wired-16-6",
                     "releaseType": "beta",
-                    "shortName": "MX 16.7"
+                    "shortName": "MX 16.6"
                 }
             ]
         },
         "cellularGateway": {
             "currentVersion": {
                 "id": 1909,
                 "firmware": "Custom version (520ac28)",
```

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

ORGANIZATION:
48[redacted]

Scan Info

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

Detailed Compliance Info

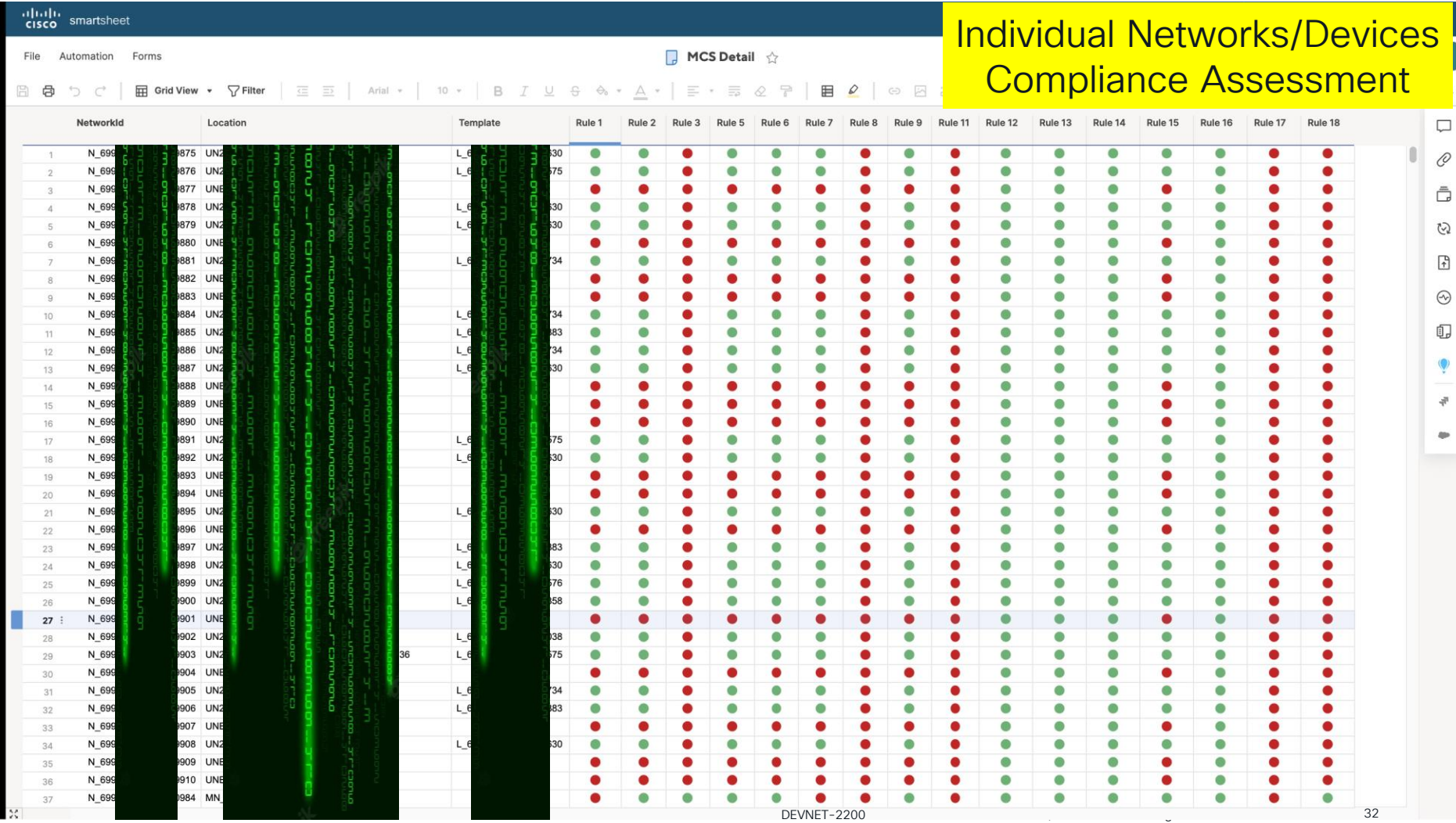
CLICK HERE

77.1%	77.8%	29.9%	78.0%	78.0%	50.7%	0.0%	79.7%	27.7%	100.0%
Rule netw_policy001 2670 / 3461 compliant	Rule netw_policy002 2694 / 3461 compliant	Rule netw_policy003 1035 / 3461 compliant	Rule netw_policy005 2699 / 3461 compliant	Rule netw_policy006 2699 / 3461 compliant	Rule netw_policy007 1754 / 3461 compliant	Rule netw_policy008 1 / 3461 compliant	Rule netw_policy009 2758 / 3461 compliant	Rule netw_policy011 958 / 3461 compliant	Rule netw_policy012 3461 / 3461 compliant
	100.0%	100.0%	77.5%	100.0%	0.0%	2.1%			
	Rule netw_policy013 3461 / 3461 compliant	Rule netw_policy014 3461 / 3461 compliant	Rule netw_policy015 2682 / 3461 compliant	Rule netw_policy016 3460 / 3461 compliant	Rule netw_policy017 0 / 3461 compliant	Rule netw_policy018 74 / 3461 compliant			

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

*not part of code repo

Individual Networks/Devices Compliance Assessment



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

Fill out your session surveys!

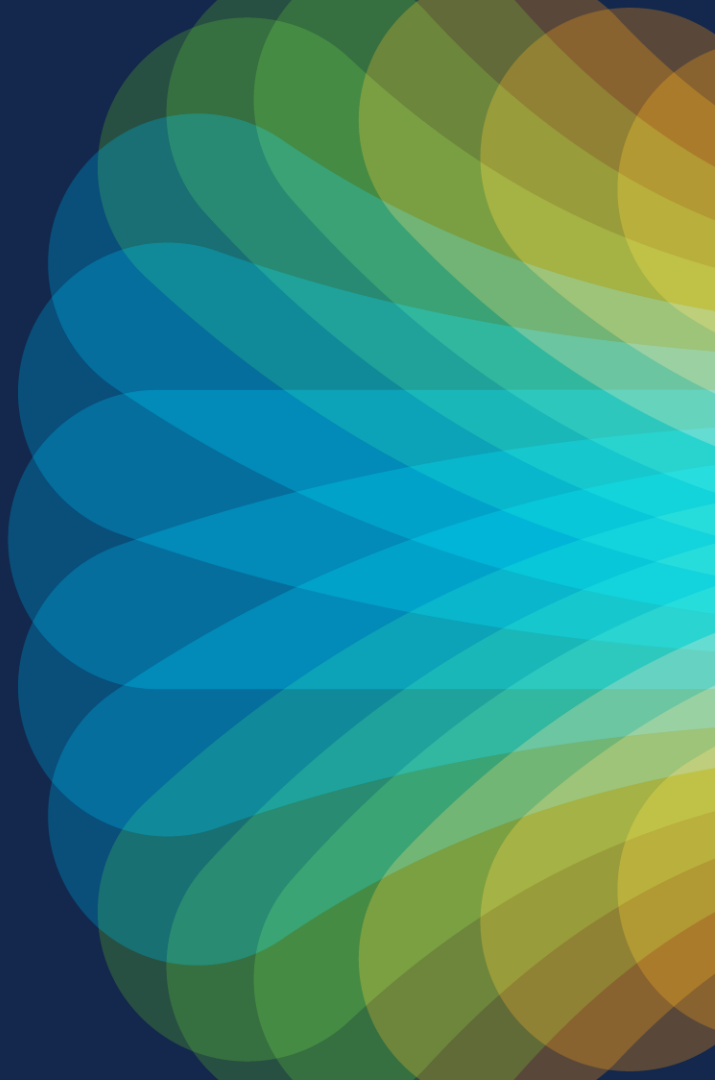
Attendees who fill out a minimum of four session surveys and the overall event survey will get **Cisco Live-branded socks** (while supplies last)!

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!
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: