



You make **possible**



API driven Internet route monitoring

with Crosswork Network Insights

Krishnan Thirukonda
Technical Marketing Engineer

DEVNET-2722

CISCO *Live!*

Barcelona | January 27-31, 2020



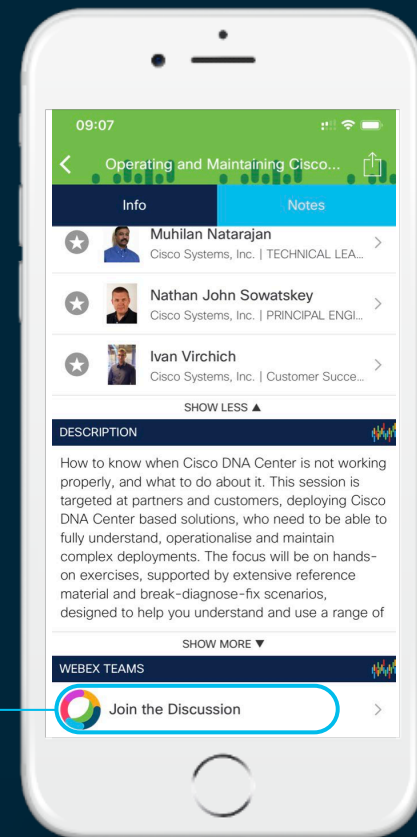
Cisco Webex Teams

Questions?

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

How

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



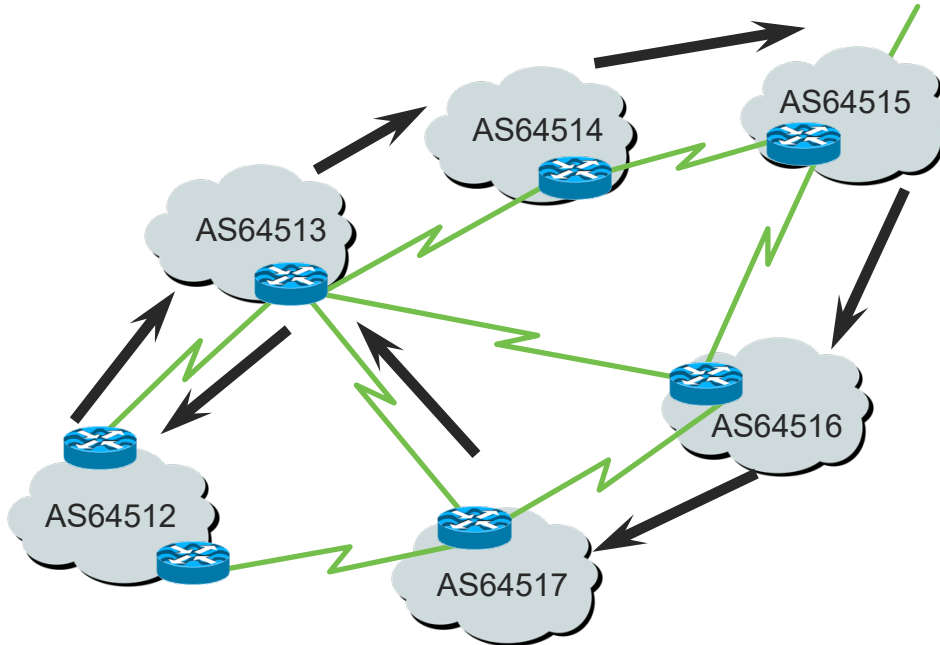
Agenda

- Introduction
 - Internet route monitoring – why do you need it
- Crosswork network insights route monitoring
- Using APIs
 - Configuring prefixes
 - Configuring ASNs
 - Configuring Endpoints
 - Configuring policies
 - Getting alarms

Why do we need Internet Route monitoring

BGP on the Internet

- On the public Internet, route distribution occurs by learning routes from a neighbor and advertising to other neighbors



Bad things happen

- Route hijack – some one advertises our prefix, and diverts all our traffic
 - Can be effectively a denial of service attack
 - Can be **malicious/intentional** or **by error/unintentional**
- **Route leak** – **someone** advertises a wrong path/prefix unintentionally
 - Traffic goes through suboptimal route or gets black-holed
- Route hijack as an attack vector
 - Man in the middle attack – some one inserts themselves in the path
 - Reconnaissance Gain information by observing traffic patterns, brute force attacks
 - Stealing cryptocurrencies
 - Steal others IP/Masquerade
 - Commit Fraud by sourcing traffic from valid Prefix/ASNs belonging to others
 - DDoS
 - Spam

The prevalence of routing incidents

[Google outage November 13, 2018](#)

"FOR TWO HOURS Monday, internet traffic that was supposed to route through Google's Cloud Platform instead found itself in quite unexpected places, including Russia and China....

Indeed, on Tuesday morning Main One said in a [statement](#) that, "This was an error during a planned network upgrade due to a misconfiguration on our BGP filters. The error was corrected within 74mins.

*Source: Wired

[Google causes outage in Japan: Aug. 26, 2017](#)

"Yesterday some Internet users would have seen issues with their Internet connectivity, experiencing slowness or parts of the Internet as unreachable. This incident hit users in Japan particularly hard and it caused the [Internal Affairs and Communications Ministry of Japan to start an investigation](#) into what caused the large-scale internet disruption that slowed or blocked access to websites and online services for dozens of Japanese companies."

*Source: BGPmon Blog

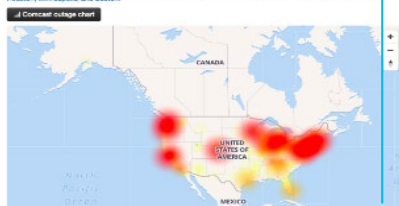
[Nationwide Comcast Outage: Nov. 6, 2017](#)

"The culprit was a configuration issue from Level 3, a telecommunications and internet service provider owned by CenturyLink. In a statement to CNN Tech, CenturyLink said a "configuration error" disrupted service and technicians restored service in 90 minutes." *

*Source: CNN

Comcast outage map

Recent reports mostly originate from: Houston, Texas; Dallas, Texas; Chicago, Illinois; New York, New York; San Francisco, California; Minneapolis, Minnesota; and Boston, Massachusetts.



cisco *Live!*

[Amazon Route 53 Outage: April. 24, 2018](#)

ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS

BORDER GATEWAY PROTOCOL ATTACK —

Suspicious event hijacks Amazon t for 2 hours, steals cryptocurrency

Almost 1,300 addresses for Amazon Route 53 rerouted for two hours.

DAN GOODIN - 4/24/2018, 3:00 PM

Verizon Route Leaks June, 2019

An Internet Service Provider in Pennsylvania (AS33154 - DQE Communications) was using a BGP optimizer in their network, which meant there were a lot of more specific **routes** in their network. ... These **routes** were supposedly "better" because they were more granular, more specific. The **leak** should have stopped at **Verizon**.

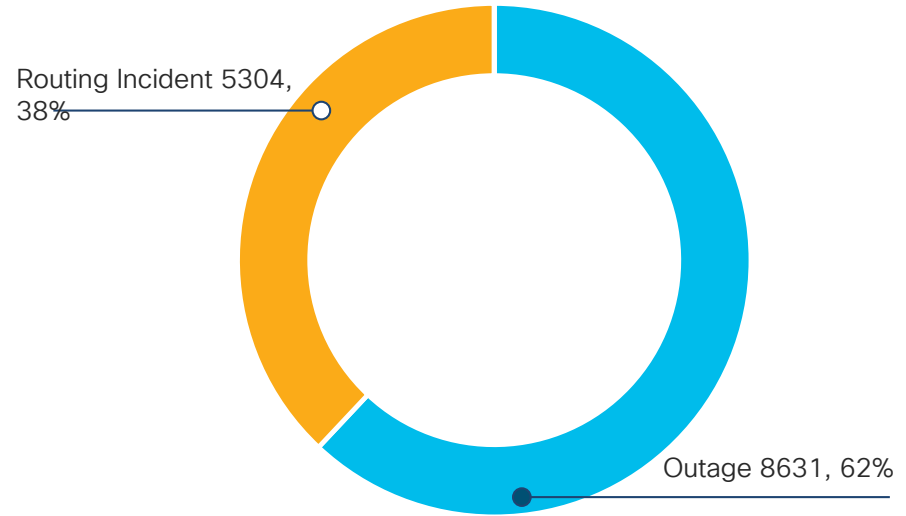
*Source: ArsTechnica



How prevalent are routing incidents?

State of Internet's routing system in 2017

- 13,935 total incidents (outages or attacks like route leaks and hijacks)
- Over 10% of all Autonomous Systems on the Internet were affected
- 3,106 Autonomous Systems were a victim of at least one routing incident
- 1,546 networks caused at least one incident



Source: [The Internet Society: 14,000 Incidents: A 2017 Routing Security Year in Review](#)

BGP on the Internet

- Historically it has been trust based – we advertise our prefixes and expect everyone to do same.
 - If we catch some one advertising wrong prefixes, we tell them not to. If it was a mistake they would comply.
 - If they do not stop advertising wrong prefixes, call their upstream providers and tell them to not accept/filter out.
- Today impacts are too high when outage occurs
- Need to monitor prefixes and ASNs belonging to yourself, your customers and critical infrastructure to make sure nothing bad is happening and remediate when bad things happen.

Mitigating routing incidents

- Routing incidents may be transient, lasting from minutes to days or weeks. Incidents may be localized.
- Often a reactive approach, post customer complaint, detecting service outage or high latency. Many incidents may go undetected.
- Traditionally, troubleshooting and verification of BGP advertisement involves use of “Looking Glass” and “Route Servers” in diverse geographical locations.
- Remediation is often an iterative process.

Challenges in Monitoring your own prefixes

Multiple diverse peers

Routing view from large number of Internet peers

Development Effort

Expertise developing analytics capabilities

Resources

Extensive Compute and Storage to maintain dataset.

Investment

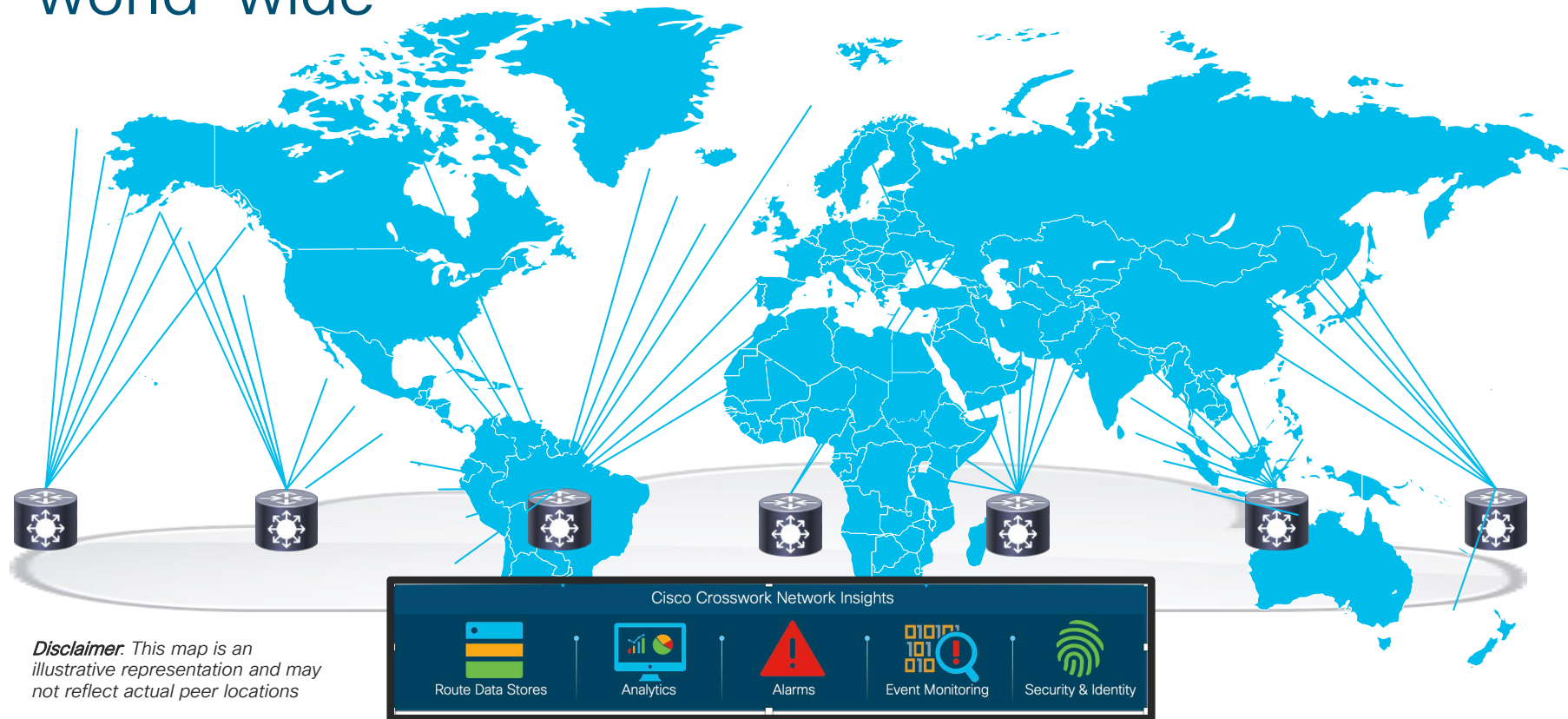
Homegrown solution requires investment and continued maintenance.

Crosswork Network Insights

“Crosswork Network Insights continuously monitors the Internet and provide real-time alerts as anomalies are detected”

Easy to use Cisco cloud based subscription service

Network insights monitors hundreds of peers world-wide





Dashboard

Alarms

ASNs

Prefixes

Policies

Settings

Help & Support

Active Alarms

Policy violations occurring now

[Manage Alarms](#)

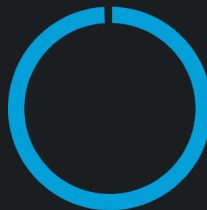
VIEW	TRIGGER	POLICY	RULE	# PEERS	SEVERITY	ACTIVATED	
View	109: Cisco Systems, Inc.	Express_109_1-migrate-asn	Unexpected AS Prefi:	10	High 	1/10/2020 11:16:17 AM	1
View	36692: Cisco Umbrella	Express_36692_1-migrate-a	Unexpected AS Prefi:	10	High 	1/10/2020 11:16:05 AM	2
View	114.29.192.0/19	Express_13445_1	SubPrefix Advertiserr	145	High 	1/10/2020 11:09:21 AM	3
View	2001:420:5000::/39	Express_109_1	Prefix Withdrawal	2	High 	1/10/2020 11:08:38 AM	4

Viewing 1 - 4 of 120 Records

[<](#) [<<](#) Page 1 of 30 [>>](#) [>](#)

Active Alarms By Rule

- 107 Prefix Withdrawal
- 11 SubPrefix Advertisement
- 2 Unexpected AS Prefix



Prefix Usage

- 117 Used
- 0 Available

Quick Jump

ASN or Prefix

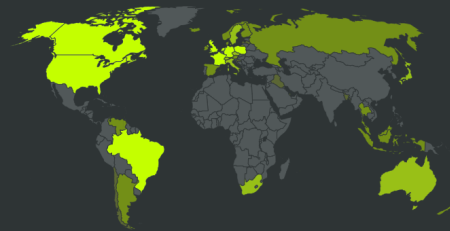
[Go](#)

Enter an ASN or prefix to view associated BGP

Application Shortcuts

- [Configure Notification Endpoints](#)
- [Express Setup](#)
- [Manage Users](#)

Violation Peers



COUNTRY

PEER COUNT

[<](#) [<<](#) [>>](#) [>](#)

API use cases with Crosswork Network Insights

Main use cases for using APIs – 1

- Administer
 - Manage (add/del/change) endpoints, users etc. (Need UI for some of this)
- Configure
 - Add/del/modify prefixes, ASNs, policies and endpoints
 - Automate for scenarios where changes happen often.
 - Hosters w Bring your own IP
 - Large orgs and Enterprises with Cloud work loads using provider IP
 - Monitor external critical infra – root dns servers etc

Main use cases for using APIs -2

- Alarms
 - Integrate with SIEM systems,
 - drive other tools say for auto-remediation – for ex: on detection of a prefix hijack, push configurations to filter or advertise more specifics etc
- Get additional insights
 - prefix attributes, (ASPATH at diff locations etc)
 - Query presence at routers – help decide if enough redundancy exists
 - More functionality coming – peer prospect analysis etc

How to use APIs with Crosswork Network Insights

Using Crosswork Network Insights

- Buy a subscription
 - <https://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/crosswork-network-automation/datasheet-c78-740228.html>
- Login to <https://crosswork.cisco.com> (using cisco.com account)
- Create API Keys
- Use API Keys to drive this service
 - Add/delete prefixes to monitor
 - Add/delete ASNs to monitor
 - Add/delete endpoints to send notifications
 - Configure monitoring policies
 - Get current alarms



Dashboard

Alarms

ASNs

Prefixes

Policies

System

Users

Endpoints

Import / Export

Settings

Help & Support

cni.user.100@gmail.com

Enabled

Edit

Role
Admin

Provider
OneID

Timezone
Auto

Theme
Dark (default)

Add API Key

Quick Search



NAME



DESCRIPTION

VALID FROM

VALID UNTIL

KEY ID



No Data

If filters are applied, try editing them

CrossworkCloud
Settings > Users > cni.demo.100@gmail.com > Help with this page

CNI DEMO CD

System: cni.demo.100@gmail.com
Users: Enabled
Endpoints:
Inp...

Dashboard
Alarms
ASNs
Prefixes
Policies

Add API Key
Used for authorizing the use of Crossword Cloud API

Name
ciscoliveapikey

Description

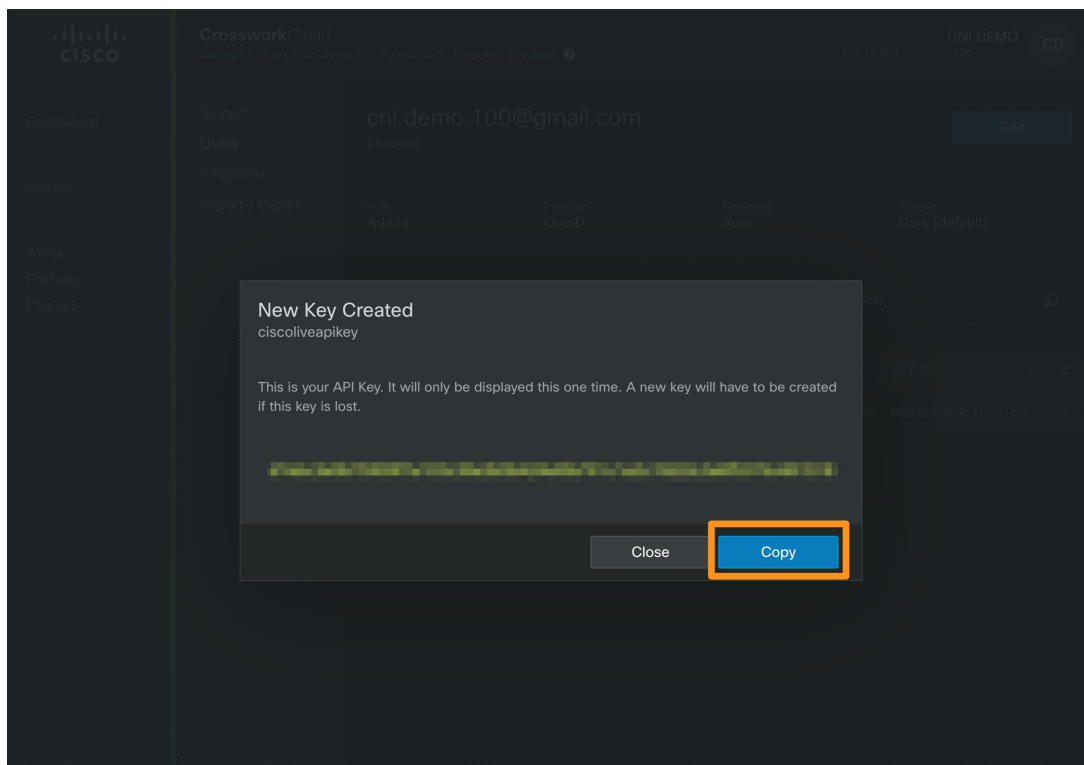
Valid
1/10/2020 12:00 AM - 2/8/2020 11:59 PM

Cancel Save

Settings
Help & Support

Viewing 0 of 0 Records

Page 0 of 0



Generate API Key here, configure lifetime needed
Protect your API Key as you would a password.

API Key

The Network Insights API Key consists of:

- An **API Key**, which is a hex encoded, 32-byte (32 hexadecimal characters) symmetric key. Client applications use the API Key to sign REST API requests destined for Network Insights.
- An **API Key identifier (ID)**, which is a unique value (32 hexadecimal characters) for the key and must be included with each signed request. Network Insights services use the Key ID to retrieve a copy of the API Key to verify the incoming request.
- API Keys should be treated securely, just like a password. Do not disclose your API Keys and ensure that they are stored securely.

Note: **API Keys have a tenant/organization scope**. A separate key will need to be used per tenant/organization that the user has access to.

Key and Keyid store

Store Key in a secure place only readable by personnel that need it - it's a secret. API Scripts need to use it.

For practice, store in the following place:

- Keyid: `~/.ccni-api-keyid.$CCNIENV`
- Key: `~/.ccni-api-key.$CCNIENV`
- Set Environment variable CCNIENV to devnet
- This allows handling different Crosswork Network insights accounts.
- CCNIENV=devnet # for lab
- CCNIENV=<production> # for production accounts

Clients use the API Key to sign all requests sent to Network Insights.

Clients need to include in the HTTP request header :

- The request signature
- The API Key ID
- Metadata detailing the fields used to determine the signature

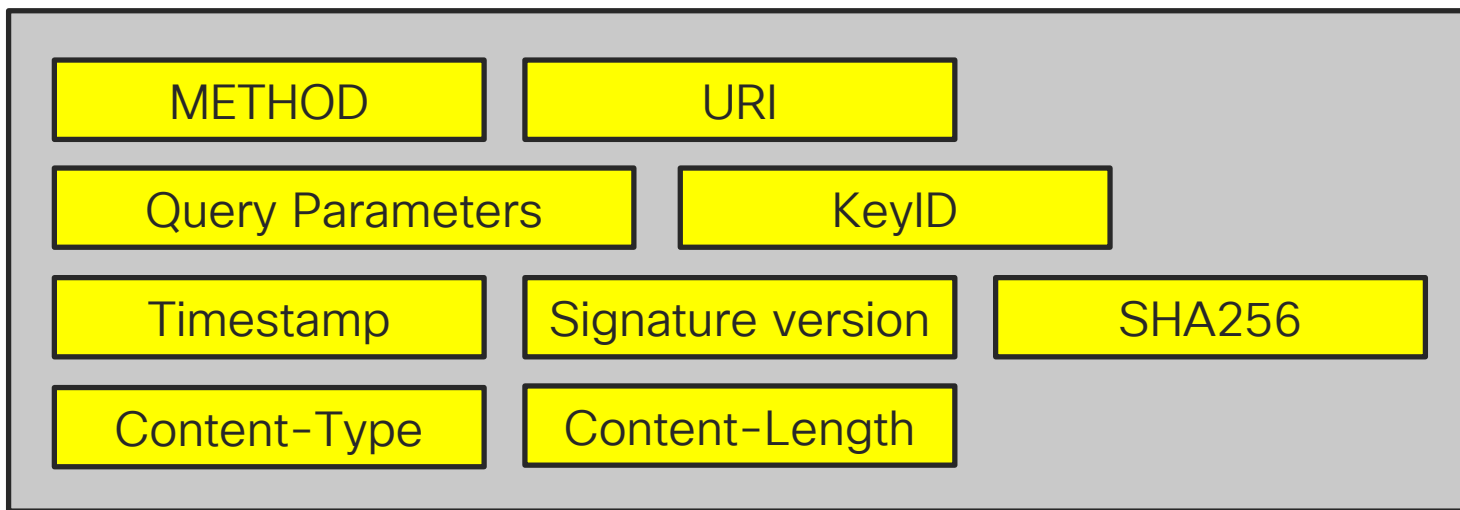
keyid

signature

```
{'User-Agent': 'python-requests/2.22.0', 'Accept-Encoding':  
'gzip, deflate', 'Accept': '*/*', 'Connection': 'keep-alive',  
'Authorization': 'hmac  
579b4b1f2aa2302e32bb2421186f0fcb:1330088da4800bf25afbe70ca0c5  
a1ff46b2e014b58d1a4b762f010e306c41ca', 'Timestamp': '2020-01-  
10T12:27:06-05:00', 'Content-Type': 'application/json', 'X-  
Cisco-Crosswork-Cloud-Signature-Version': '1.0'}
```

Request Signing

- Generating Signature for request



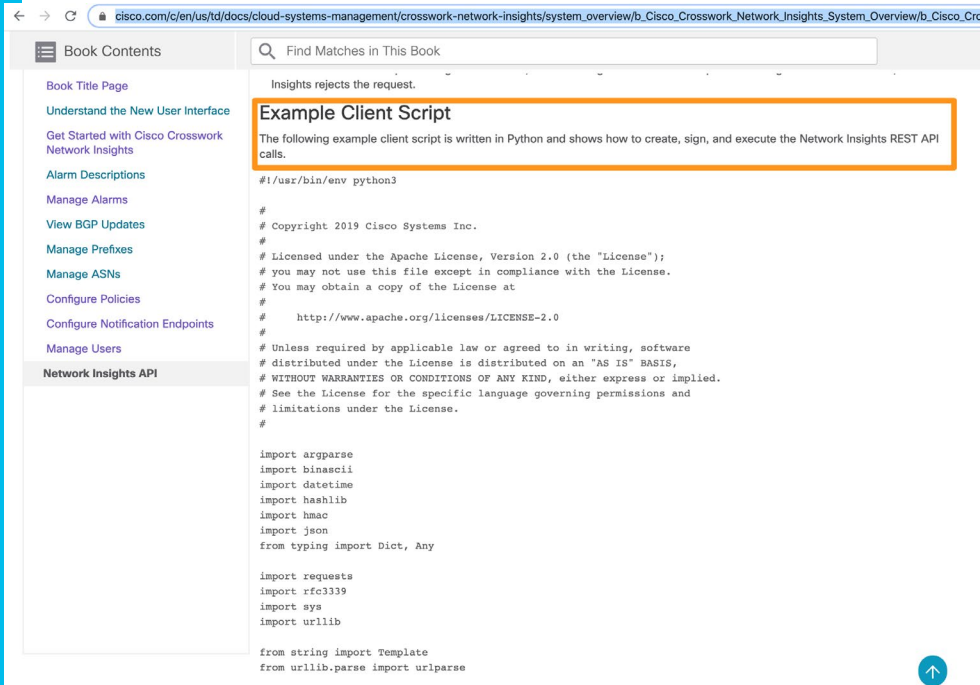
Network Insights performs the following steps:

1. Extracts the requested parameters.
2. Uses the API Key ID to retrieve the API Key and associated metadata.
3. Recalculates the signature.
4. Compares the calculated signature with the requested signature.
5. If the calculated and requested signatures match, Network Insights forwards the request. If the signatures do not match, Network Insights rejects the request.

Use python script in our doc to get started

https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/crosswork-network-insights/system_overview/b_Cisco_Crosswork_Network_Insights_System_Overview/b_Cisco_Crosswork_Network_Insights_System_Overview_chapter_01011.html

Copy this to ~/crosswork.py



The screenshot shows a web browser displaying a Cisco documentation page. The URL in the address bar is https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/crosswork-network-insights/system_overview/b_Cisco_Crosswork_Network_Insights_System_Overview/b_Cisco_Crosswork_Network_Insights_System_Overview_chapter_01011.html. The page has a sidebar with a 'Book Contents' menu. The main content area is titled 'Example Client Script' and contains a Python script for interacting with the Network Insights REST API. The script includes a license notice and imports various modules like argparse, binascii, datetime, hashlib, hmac, json, and requests.

Book Contents

- Book Title Page
- Understand the New User Interface
- Get Started with Cisco Crosswork Network Insights
- Alarm Descriptions
- Manage Alarms
- View BGP Updates
- Manage Prefixes
- Manage ASNs
- Configure Policies
- Configure Notification Endpoints
- Manage Users
- Network Insights API

Find Matches in This Book

Insights rejects the request.

Example Client Script

The following example client script is written in Python and shows how to create, sign, and execute the Network Insights REST API calls.

```
#!/usr/bin/env python3

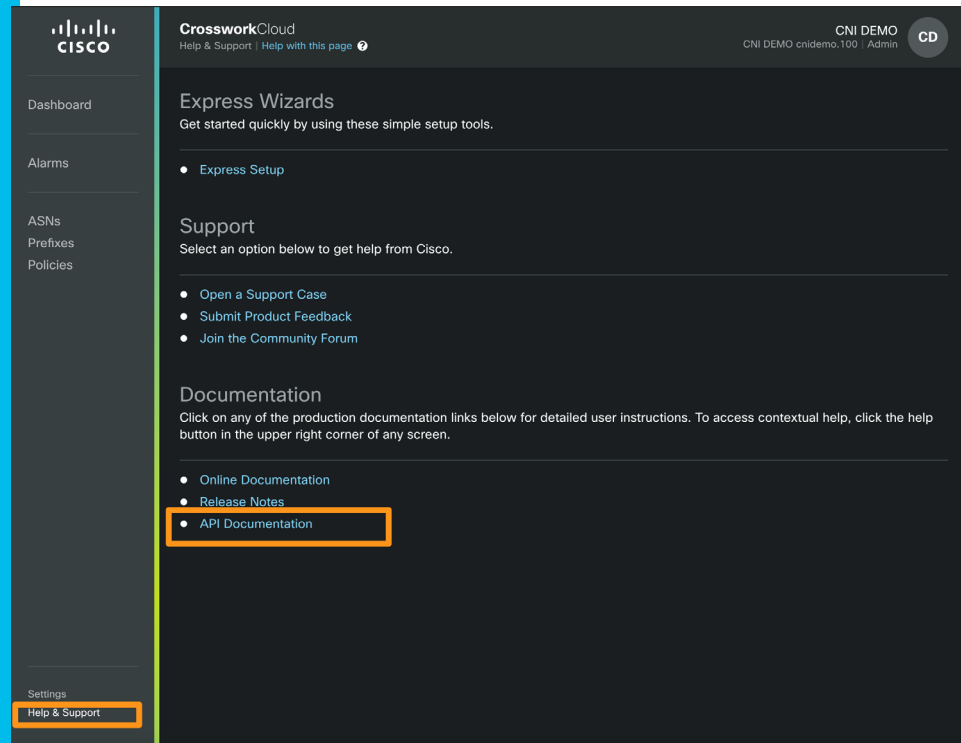
#
# Copyright 2019 Cisco Systems Inc.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

import argparse
import binascii
import datetime
import hashlib
import hmac
import json
from typing import Dict, Any

import requests
import rfc3339
import sys
import urllib

from string import Template
from urllib.parse import urlparse
```

API Details can be accessed from within Crosswork Network Insights



Json payloads schema documented

URI	Method	Schema	Description
/api/beta/alarms/{alarm_id}/policies	GET	response	Returns a list of configured policies for a specific alarm. Note that this will only return data if the named alarm has been triggered.
/api/beta/asns	GET	response	Returns a list of all ASNs.
/api/beta/asns/{asn}	GET	response	Returns details on a specific ASN.
/api/beta/config	GET	response	Download the current configuration.
/api/beta/config	POST	response request	Upload a configuration to the service. Note: This call overwrites any existing configuration. It uploads a new configuration and can be used to bootstrap an initial configuration. Use the <i>/api/beta/provision</i> REST call for specific configuration requests.
/api/beta/config/epoch/{epoch}	GET	response	Returns the configuration for a specific epoch. Enter an epoch number to return the configuration corresponding to that specific epoch. An epoch value of 0 returns the configuration for the most recent epoch.
/api/beta/notifications/endpoints	GET	response	Returns a list of all configured notification endpoints.
/api/beta/notifications/endpoints/{euuid}	GET	response	Returns details on a specific notification endpoint.
/api/beta/policies	GET	response	Returns a list of all configured policies.
/api/beta/policies/{policy_id}	GET	response	Returns details on a specific configured policy.
/api/beta/policies/{policy_id}/alarms	GET	response	Returns a list of alarms for a specific configured policy.
/api/beta/prefixes	GET	response	Returns a list of prefixes to which an organization is subscribed.

Response JSON Schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "properties": {
    "error": {
      "properties": {
        "causes": {
          "items": {
            "type": "string"
          },
          "type": "array"
        },
        "code": {
          "type": "integer"
        },
        "critical": {
          "type": "boolean"
        }
      }
    }
  }
}
```

Response JSON Example

```
{
  "prefixes": [
    {
      "prefix": "1.1.1.1/32",
      "lastModifiedTs": "1566436610598"
    },
    {
      "prefix": "2.2.2.2/32",
      "lastModifiedTs": "1566436610598"
    },
    {
      "prefix": "2001:30:101::/48",
      "lastModifiedTs": "1566436610652"
    },
    {
      "prefix": "2001:30:101::/48"
    }
  ]
}
```


Shortcuts GET with python script for CCNI API

```
# For demo today we will use a short cut that's handy
# This is GET short cut, echoes the cmd and then runs it

(crosswork-api) CSC-M-C4WX:devnet csc$ cat ./ccniget.sh
echo ~/crosswork.py --keyid cat ~/.ccni-api-keyid.$CCNIENV --key
cat ~/.ccni-api-key.$CCNIENV --host crosswork.cisco.com --method
GET --uri /api/beta/$*

echo +++

~/crosswork.py --keyid $(cat ~/.ccni-api-keyid.$CCNIENV) --key
$(cat ~/.ccni-api-key.$CCNIENV) --host crosswork.cisco.com --
method GET --uri /api/beta/$*

echo ---
```

CCNI API GET examples

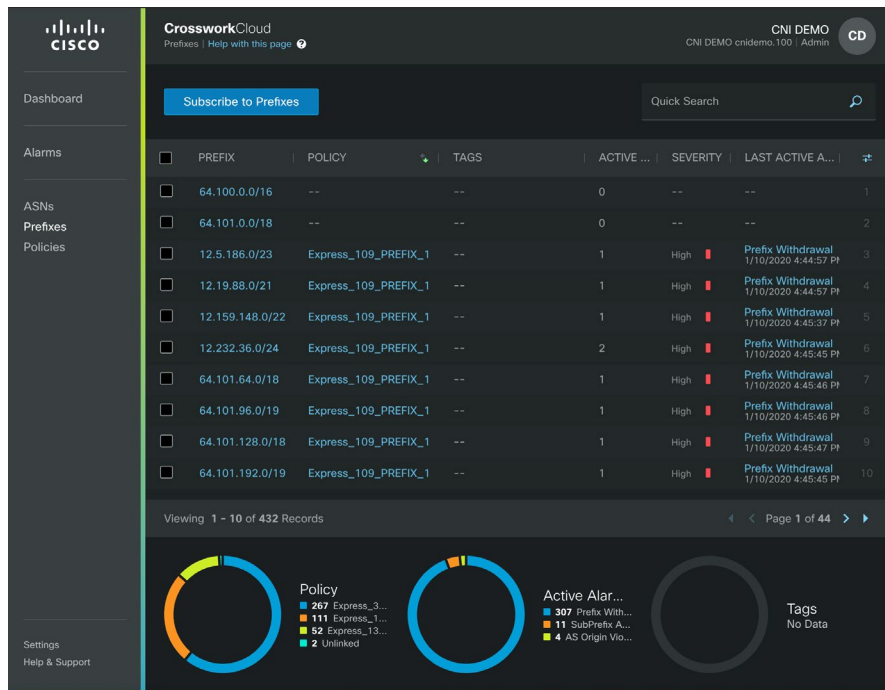
```
(crosswork-api) CSC-M-C4WX:devnet csc$ ./ccniget.sh prefixes
```

```
/Users/csc/crosswork.py --keyid cat /Users/csc/.ccni-api-keyid.devnet --key cat /Users/csc/.ccni-api-key.devnet --host crosswork.cisco.com --method GET --uri /api/beta/prefixes
```

```
+++
```

```
{  
  "prefixes": [  
    {  
      "prefix": "64.100.0.0/16",  
      "lastModifiedTs": "1578594930055"  
    },  
    {  
      "prefix": "64.101.0.0/18",  
      "lastModifiedTs": "1578594930055"  
    }  
  ]  
}
```

cisco *Live!*



Provision python script for CCNI API

```
# This is POST to /api/beta/provision short cut, prints the payload and then posts it

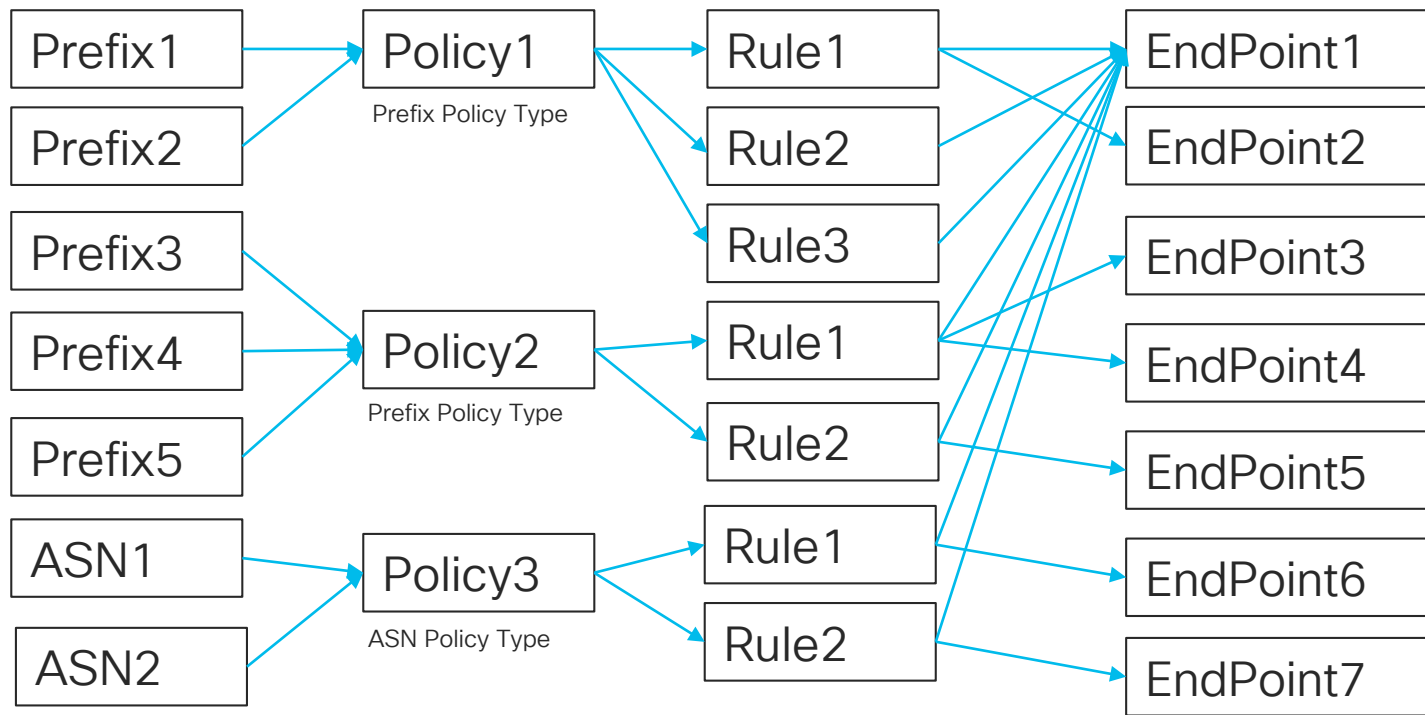
# provision api is used to make create/update/deletes

(crosswork-api) CSC-M-C4WX:devnet csc$ cat ./ccniprovision.sh
echo ~/crosswork.py --keyid cat ~/.ccni-api-keyid.$CCNIENV --key cat ~/.ccni-api-
key.$CCNIENV --host crosswork.cisco.com --method POST --uri /api/beta/provision --
payload $1
echo "$1:"
cat $1
echo +++
~/crosswork.py --keyid $(cat ~/.ccni-api-keyid.$CCNIENV) --key $(cat ~/.ccni-api-
key.$CCNIENV) --host crosswork.cisco.com --method POST --uri /api/beta/provision --
payload $1
echo ---
```

Each Prefix/ASN maps to one policy.

Policy has rules

Each rule can map to any number of endpoints



Supported Alarm Types

	Supported
AS Origin Violation	✓
SubPrefix Advertisement	✓
Prefix Withdrawal	✓
ROA Failure	✓
Upstream AS Change	✓
Parent Aggregate Change	✓
Unexpected AS Prefix	✓
AS Path Length Violation	✓
Prefix Advertisement	✓
Valid AS Path Violation	✓

Supported Notification Endpoints



Email



SMS



Amazon S3



Slack



WebEx Teams

Recommended Provisioning sequence

- Add endpoints that will receive notifications
 - Get endpoint ids
- Create Policies with rules and map to endpoints that need to be notified
 - Create Prefix policies, will need policy_ids for adding to prefixes
 - Create ASN policies, will need policy_ids for adding to ASNs
- Add prefixes
 - Use policy_ids for applying for the specific policy needed for this policy
- Add ASNs
 - Use policy_ids for applying the specific policy needed for this ASN

Provision notification end point with CCNI API

```
(crosswork-api) CSC-M-C4WX:devnet csc$ ./ccniprovision.sh set-endpoint.json
/Users/csc/crosswork.py --keyid cat /Users/csc/.ccni-api-keyid.devnet --key cat /Users/csc/.ccni-api-
key.devnet --host crosswork.cisco.com --method POST --uri /api/beta/provision --payload set-endpoint.json
set-endpoint.json:
```

```
{
  "operations":
  [
    {
      "set_notification_endpoint_request": {
        "email": {
          "email": "cni.demo.100@gmail.com"
        },
        "name": "demonotification"
      }
    }
  ]
}
+++
{
  "results": [
    {
      "setNotificationEndpointResponse": {
        "euuid": "253e5021-8676-4c12-8d56-2e7353bfbda4"
        "sendConfirmation": true,
        "confirmationStatus": "Sent email to cni.demo.1
      }
    }
  ]
}
---
```

The screenshot shows the CrossworkCloud interface. On the left is a sidebar with navigation links: Dashboard, Alarms, ASNs, Prefixes, Policies, Settings, and Help & Support. The main content area is titled 'CrossworkCloud' and 'CNI DEMO'. It has tabs for 'Email', 'SMS', 'Slack', 'Webex Teams', and 'Amazon S3'. A 'Create Email Endpoint' button is visible. Below it is a table with columns: NAME, EMAIL, VERIFIED, STATUS, and an icon column. The table contains two rows: 'demonotification' and 'mynotificationendpoint'. The 'demonotification' row has a status of 'Disabled' and a verification of 'No'. The 'mynotificationendpoint' row has a status of 'Disabled' and a verification of 'Yes'. At the bottom, it says 'Viewing 1 - 2 of 2 Records'.

NAME	EMAIL	VERIFIED	STATUS	
demonotification	cni.demo.100@gmail.com	No	Disabled	1
mynotificationendpoint		Yes	Disabled	2

Provision prefixes with CCNI API

```
(crosswork-api) CSC-M-C4WX:devnet csc$ ./ccniprovision.sh set-prefix-request.json
/Users/csc/crosswork.py --keyid cat /Users/csc/.ccni-api-keyid.devnet --key cat /Users/csc/.ccni-api-
key.devnet --host crosswork.cisco.com --method POST --uri /api/beta/provision --payload set-prefix-
request.json
set-prefix-request.json:
{
  "operations": [
    {
      "setPrefixRequest": {
        "prefix": "64.100.0.0/16", "tags" : [ "devnet", "lab" ]
      },
      "o_creat": true,
      "o_excl": true
    },
    {
      "setPrefixRequest": {
        "prefix": "64.101.0.0/18", "tags" : [ "devnet", "lab" ]
      },
      "o_creat": true,
      "o_excl": true
    }
  ]
}
+++
```

Provision prefixes with CCNI API – contd...

```
+++
{
  "results": [
    {
      "setPrefixResponse": {
        "prefix": "64.100.0.0/16"
      }
    },
    {
      "setPrefixResponse": {
        "prefix": "64.101.0.0/18"
      }
    }
  ]
}
(crosswork-api) CSC-M-C4WX:devnet csc$
```

CrossworkCloud
Prefixes | Help with this page

Krishnan Thirukonda
CNI DEMO cndemo.100 | Admin

Subscribe to Prefixes

Quick Search

<input type="checkbox"/>	PREFIX	POLICY	TAGS	ACTIVE ALA...	SEVERITY	LAST ACTIVE ALARM	
<input type="checkbox"/>	64.100.0.0/16	--	devnet lab	0	--	--	1
<input type="checkbox"/>	64.101.0.0/18	--	devnet lab	0	--	--	2

Viewing 1 - 2 of 2 Records

Policy
■ 2 Unlinked

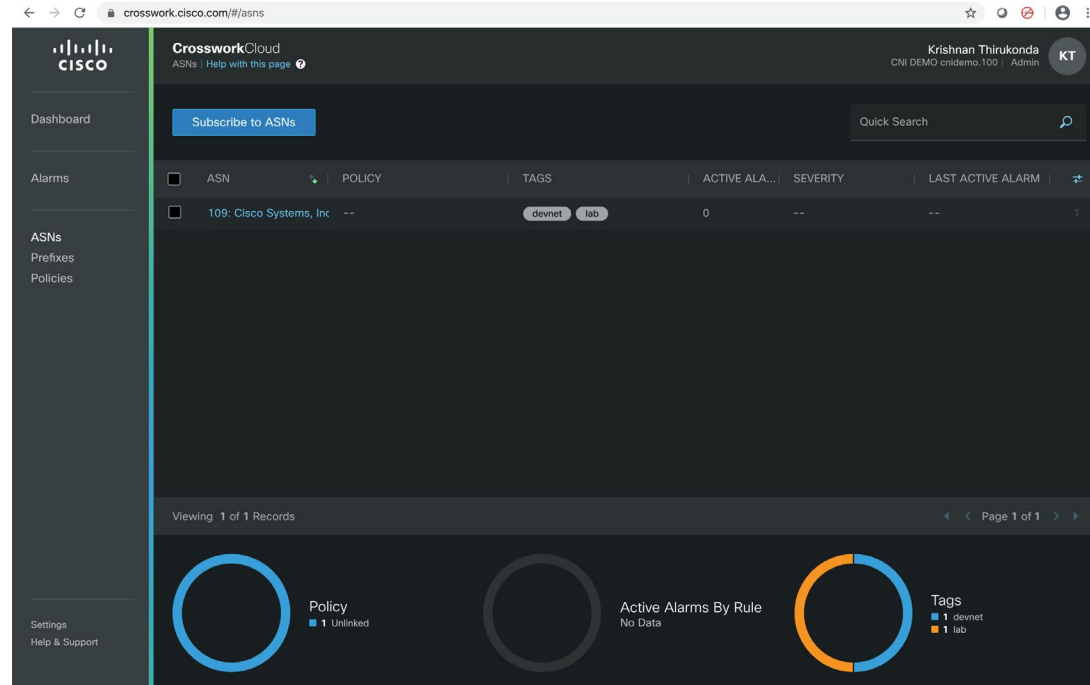
Active Alarms By Rule
No Data

Tags
■ 2 devnet
■ 2 lab

Provision ASNs with CCNI API

```
(crosswork-api) CSC-M-C4WX:devnet csc$ ./ccniprovision.sh set-asn.json
/Users/csc/crosswork.py --keyid cat /Users/csc/.ccni-api-keyid.devnet --key cat /Users/csc/.ccni-api-key.devnet --host crosswork.cisco.com --method POST --uri /api/beta/provision --payload set-asn.json
set-asn.json:
```

```
{
  "operations": [
    {
      "set_asn_request": {
        "asn": "109",
        "tags": ["devnet", "lab"]
      },
      "o_creat": true,
      "o_excl": true
    }
  ]
}
+++
{
  "results": [
    {
      "setAsnResponse": {
        "asn": 109
      }
    }
  ]
}
---
```



Provision policy with CCNI API

```
(crosswork-api) CSC-M-C4WX:devnet csc$ ./ccniprovision.sh set-policy.json
/Users/csc/crosswork.py --keyid cat /Users/csc/.ccni-api-keyid.devnet --key cat /Users/csc/.ccni-api-
key.devnet --host crosswork.cisco.com --method POST --uri /api/beta/provision --payload set-policy.json
set-policy.json:
```

```
{
  "operations": [
    {"set_group_request": {
      "group_name": "lab-policy",
      "origin_asns": [ 109 ,7018] ,
      "policy_type": "PREFIX_POLICY" ,
      "tags": ["devnet", "lab"],
      "upstream_asns": [ 109 ,7018 ] ,
      "alarms": {
        "1" : {
          "alarm_type": "ORIGIN_ASN_VIOLATION" ,
          "endpoints": [ {
            "endpoint_ids": [ "d66356bb-486c-4e48-a879-6ce9b0a69c88" ] ,
            "max_threshold": 1 ,
            "min_threshold": 0
          }
        ],
        "level": 3 ,
        "max_peer_count": 1 ,
        "min_peer_count": 0 ,
      }
    }
  ]
  "origin_asn_violation": { }
}
}}}}
+++
```

Provision policy with CCNI API

```
+++  
{  
  "results": [  
    {  
      "setGroupResponse": {  
        "guid": "5d157e9c-3b81-4abd-aa89-d94f027b8426"  
      }  
    }  
  ]  
}  
----
```

The screenshot displays the Cisco CrossworkCloud interface for configuring a policy. The left sidebar contains navigation links: Dashboard, Alarms, ASNs, Prefixes, Policies, Settings, and Help & Support. The main content area is titled 'lab-policy' and includes buttons for 'Delete', 'Duplicate', and 'Edit'. Below the title, there are tabs for 'Overview', 'Prefixes', and 'Alarms'. The 'Overview' tab is active, showing a table with columns for 'Expected Origin ASNs', 'Expected Upstream ASNs', and 'Valid AS Path Pattern'. The table contains the following data:

Expected Origin ASNs	Expected Upstream ASNs	Valid AS Path Pattern
109	701, 7018	--

Below the table, there is a section titled 'Rules 1' with a sub-section 'AS Origin Violation' showing '0 Active Alarms'. A table below this section provides details about the violation:

Status	Severity
Enabled	High
Peers To Resolve	Peers To Trigger
0	1
Endpoints	
mynotificationendpoint Email	

Get Alarms -1

```
(crosswork-api) CSC-M-C4WX:devnet csc$ ./ccniget.sh 'alarms?state=1&maxRecords=1'
/Users/csc/crosswork.py --keyid cat /Users/csc/.ccni-api-keyid.devnet --key cat /Users/csc/.ccni-api-
key.devnet --host crosswork.cisco.com --method GET --uri /api/beta/alarms?state=1&maxRecords=1
+++
{
  "alarmData": [
    {
      "auuid": "0030c075-a582-4b21-bb85-4ccbb481167a",
      "guuid": "be2989fd-6b16-4be6-9c7b-dbe34876fe66",
      "groupName": "Express_36692_PREFIX_1",
      "alarmType": "WITHDRAWN_VIOLATION",
      "currentState": {
        "state": "ACTIVE"
      },
      "stateTransitionAt": "2020-01-10T21:45:52.565295140Z",
      "level": "RED",
      "orgUuid": "65d7d7f9-023c-43bd-8072-4fc205cc8c04",
      "alarmDetailsMsg": {
        "auuid": "0030c075-a582-4b21-bb85-4ccbb481167a",
        "at": "2020-01-10T21:45:45.094610991Z",
        "condition": "ALARM_CONDITION_ACTIVE",
        "alarm": {
          "prefixWithdrawnAlarm": {
            "configuredPrefix": {
              "ipAddrType": "IPV4",
              "length": 24,
              "address": "knBlAA=="
            },
            "configuredPrefixString": "146.112.101.0/24",

```

Get Alarms - 2

```
"configuredPrefixString": "146.112.101.0/24",
"prefixWithdrawnViolationPeers": [
  {
    "peer": "72",
    "at": "2019-09-23T20:25:40.383891293Z"
  },
  {
    "peer": "374",
    "at": "2019-12-01T07:15:48.525128415Z"
  }
]
},
"alarmType": "WITHDRAWN_VIOLATION",
"orgUuid": "65d7d7f9-023c-43bd-8072-4fc205cc8c04",
"violationPeerCount": 19,
"maxPeerCount": 5,
"lastAlarmActiveAt": "2020-01-10T21:45:45.094610991Z"
}
},
"numRecordsReturned": 1,
"opaqueToken":
"TmpWa04yUTNaamt0TURJell5MDBNMkprTFRnd056SXROR1pqTWpBMVkyTTRZekEwT2pBd016Umxav1ZtTFRVMU1qSXROR0l4WWkxaFlq
ZGlMVGRqWm1ZMk5XTTVOemsyTWc9PQ=="
}
---
(crosswork-api) CSC-M-C4WX:devnet csc$
```

CrossworkCloud

Alarms | [Help with this page](#)

CNI DEMO

CNI DEMO cnidemo.100 | Admin

CD

Dashboard

Alarms

ASNs

Prefixes

Policies

Settings

Help & Support

Active

Acknowledged

History

Quick Search

	VIEW	TRIGGER	POLICY	RULE	# PEERS	SEVERITY	ACTIVATED		
<input type="checkbox"/>	View	67.215.73.0/24	Express_36692_PRE	Prefix Withdraw	54	High 	1/10/2020 5:46:19 F		1
<input type="checkbox"/>	View	208.67.218.0/24	Express_36692_PRE	Prefix Withdraw	17	High 	1/10/2020 5:17:39 F		2
<input type="checkbox"/>	View	208.67.219.0/24	Express_36692_PRE	Prefix Withdraw	17	High 	1/10/2020 5:17:36 F		3
<input type="checkbox"/>	View	146.112.52.0/24	Express_36692_PRE	Prefix Withdraw	19	High 	1/10/2020 5:16:19 F		4
<input type="checkbox"/>	View	146.112.217.0/24	Express_36692_PRE	Prefix Withdraw	19	High 	1/10/2020 5:16:18 F		5
<input type="checkbox"/>	View	146.112.226.0/24	Express_36692_PRE	Prefix Withdraw	19	High 	1/10/2020 5:16:18 F		6
<input type="checkbox"/>	View	146.112.138.0/24	Express_36692_PRE	Prefix Withdraw	19	High 	1/10/2020 5:16:18 F		7
<input type="checkbox"/>	View	146.112.222.0/24	Express_36692_PRE	Prefix Withdraw	19	High 	1/10/2020 5:16:13 F		8
<input type="checkbox"/>	View	146.112.63.0/24	Express_36692_PRE	Prefix Withdraw	19	High 	1/10/2020 5:16:13 F		9
<input type="checkbox"/>	View	146.112.66.0/24	Express_36692_PRE	Prefix Withdraw	36	High 	1/10/2020 5:16:12 F		10
<input type="checkbox"/>	View	146.112.150.0/24	Express_36692_PRE	Prefix Withdraw	19	High 	1/10/2020 5:16:12 F		11
<input type="checkbox"/>	View	146.112.187.0/24	Express_36692_PRE	Prefix Withdraw	18	High 	1/10/2020 5:16:12 F		12
<input type="checkbox"/>	View	146.112.55.0/24	Express_36692_PRE	Prefix Withdraw	18	High 	1/10/2020 5:16:12 F		13
<input type="checkbox"/>	View	146.112.247.0/24	Express_36692_PRE	Prefix Withdraw	19	High 	1/10/2020 5:16:12 F		14
<input type="checkbox"/>	View	67.215.84.0/24	Express_36692_PRE	Prefix Withdraw	19	High 	1/10/2020 5:15:39 F		15

Viewing 1 - 15 of 322 Records

◀

<

Page 1 of 22

>

▶

Get Next Alarm

```
(crosswork-api) CSC-M-C4WX:devnet csc$ ./ccniget.sh
'alarms?state=1&maxRecords=1&opaqueToken=TmpWa04yUTNaamt0TURJell5MDBNMkprTFRnd056SXROR1pqTWpBMVkyTTRZekEw
T2pBd016UmxaV1ZtTFRVMU1qSXROR014WWkxaFlqZG1MVGRqWm1ZMk5XTTVOemsyTWc9PQ=='
{
  "alarmData": [
    {
      "auuid": "00628e14-7740-43dd-b51a-6cbe80313b84",
      "guuid": "b12742bf-d7e1-4177-95dc-47acf3195775",
      "groupName": "Express_13445_PREFIX_1",
      "alarmType": "WITHDRAWN_VIOLATION",
      "currentState": {
        "state": "ACTIVE"
      },
      "stateTransitionAt": "2020-01-10T21:46:16.703841110Z",
      "level": "RED",
      "orgUuid": "65d7d7f9-023c-43bd-8072-4fc205cc8c04",
      "alarmDetailsMsg": {
        "auuid": "00628e14-7740-43dd-b51a-6cbe80313b84",
        "at": "2020-01-10T21:45:55.687169771Z",
        "condition": "ALARM_CONDITION_ACTIVE",
        "alarm": {
          "prefixWithdrawnAlarm": {
            "configuredPrefix": {
              "ipAddrType": "IPV4",
              "length": 21,
              "address": "RRq4AA=="
            },
            "configuredPrefixString": "69.26.184.0/21",

```

Get Specific Alarm Detail – 1

```
(crosswork-api) CSC-M-C4WX:devnet csc$ ./ccniget.sh alarms/00628e14-7740-43dd-b51a-6cbe80313b84
/Users/csc/crosswork.py --keyid cat /Users/csc/.ccni-api-keyid.devnet --key cat /Users/csc/.ccni-api-
key.devnet --host crosswork.cisco.com --method GET --uri /api/beta/alarms/00628e14-7740-43dd-b51a-
6cbe80313b84
```

```
+++
```

```
{
  "auuid": "00628e14-7740-43dd-b51a-6cbe80313b84",
  "guuid": "b12742bf-d7e1-4177-95dc-47acf3195775",
  "groupName": "Express_13445_PREFIX_1",
  "alarmType": "WITHDRAWN_VIOLATION",
  "currentState": {
    "state": "ACTIVE"
  },
  "stateTransitionAt": "2020-01-10T21:46:16.703841110Z",
  "level": "RED",
  "orgUuid": "65d7d7f9-023c-43bd-8072-4fc205cc8c04",
  "alarmDetailsMsg": {
    "auuid": "00628e14-7740-43dd-b51a-6cbe80313b84",
    "at": "2020-01-10T21:45:55.687169771Z",
    "condition": "ALARM_CONDITION_ACTIVE",
    "alarm": {
      "prefixWithdrawnAlarm": {
        "configuredPrefix": {
          "ipAddrType": "IPV4",
          "length": 21,
          "address": "RRq4AA=="
        },
        "configuredPrefixString": "69.26.184.0/21",

```

Get Specific Alarm Detail -2

```
    "prefixWithdrawnViolationPeers": [
      {
        "peer": "430",
        "at": "2019-12-23T01:15:18.846610139Z"
      },
      {
        "peer": "447",
        "at": "2019-12-16T08:08:47.011974980Z"
      }
    ]
  },
  "alarmType": "WITHDRAWN_VIOLATION",
  "orgUuid": "65d7d7f9-023c-43bd-8072-4fc205cc8c04",
  "violationPeerCount": 5,
  "maxPeerCount": 5,
  "lastAlarmActiveAt": "2020-01-10T21:45:55.687169771Z"
}
}
---
(crosswork-api) CSC-M-C4WX:devnet csc$
```

Crosswork Network Insights enables monitoring your Internet Routing Assets

APIs enable automated use and auto-remediation

Complete your online session survey



- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education



Demos in the
Cisco Showcase



Walk-In Labs



Meet the Engineer
1:1 meetings



Related sessions

See us at DevNet Service Provider Booth



Thank you





You make **possible**