
cisco Live!
Let's go

The bridge to possible

ISE automation hands on

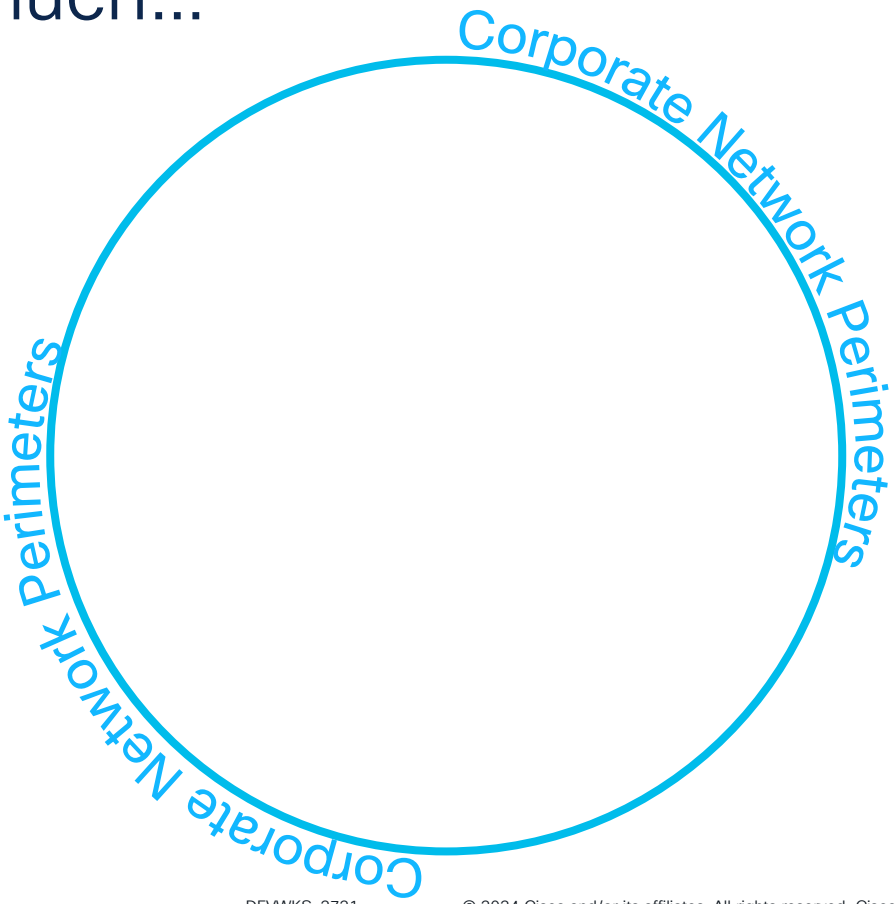
Barry Yuan, Security Technical Solutions Architect

DEVWKS-2721T

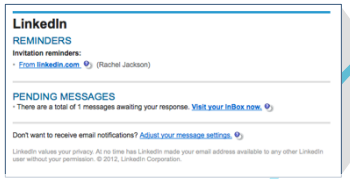
Agenda

- Introduction
- ISE use cases
- ISE API updates use cases
- Hands on lab
- Optional challenge
- Continue your education
- Conclusion

When we trust too much...



When we trust too much...

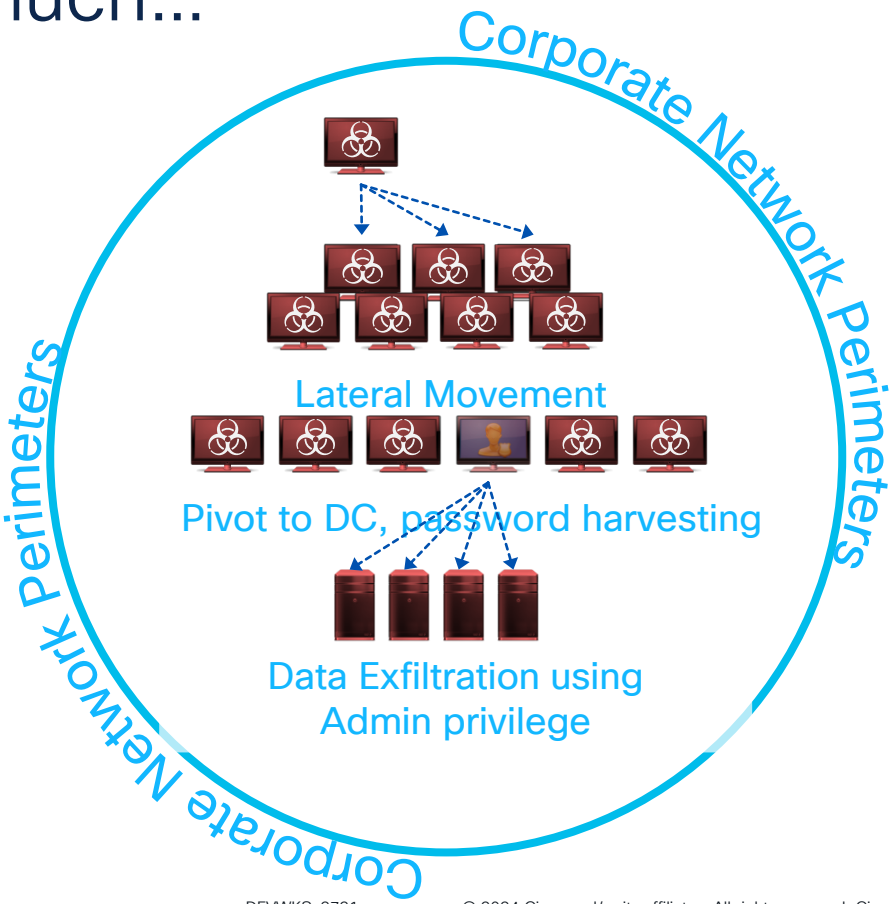


Victim clicks phishing email link, or

Corporate Network Perimeters

Corporate Network Perimeters

When we trust too much...

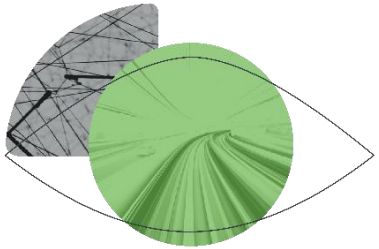


When we trust too much...

Information monetized

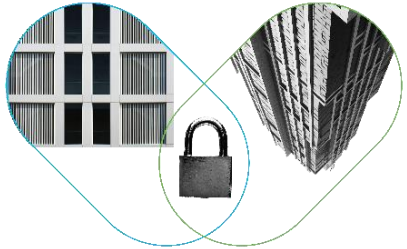
The Foundations of Zero Trust in Your Workplace

Visibility



Grant the right level of network access to users across domains

Segmentation



Shrink zones of trust and grant access based on least privilege

Containment

Automate containment of infected endpoints and revoke network access

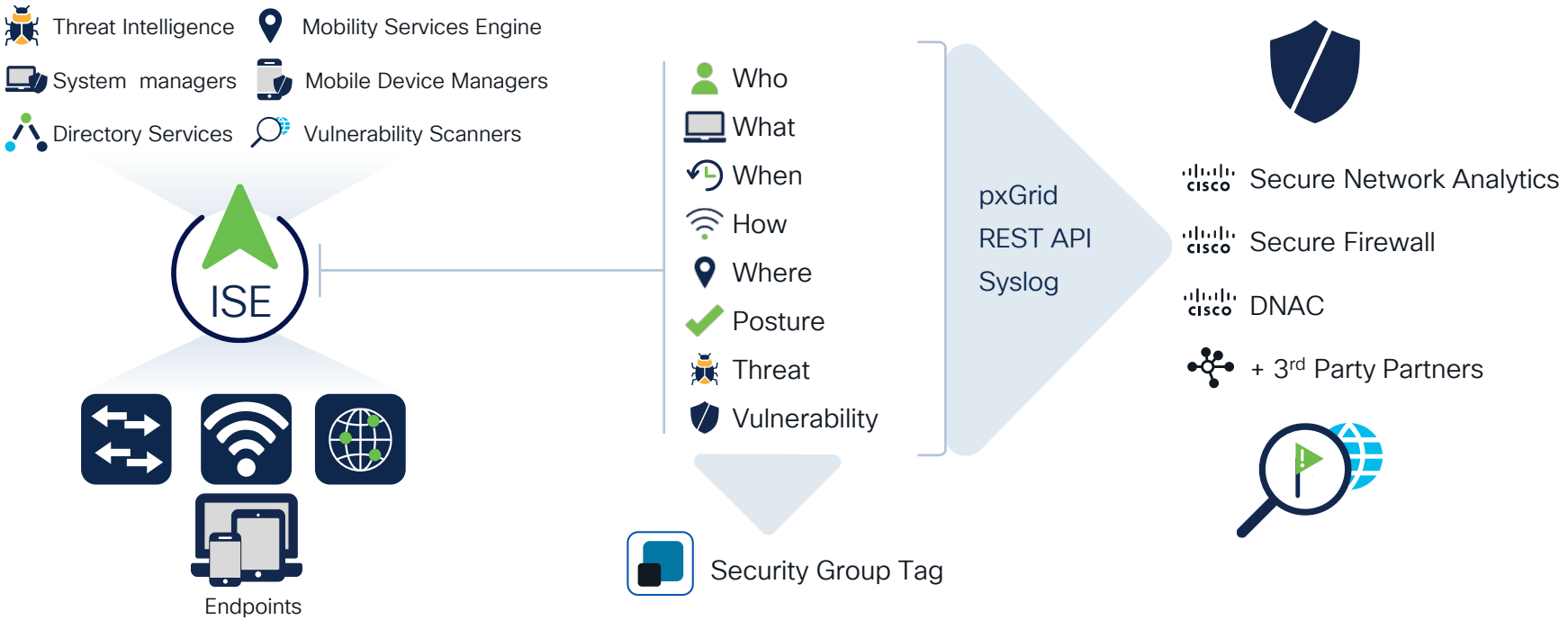
Context Build, Summarize, Exchange

Visibility and Access Control

ISE builds context and applies access control restrictions to users and devices

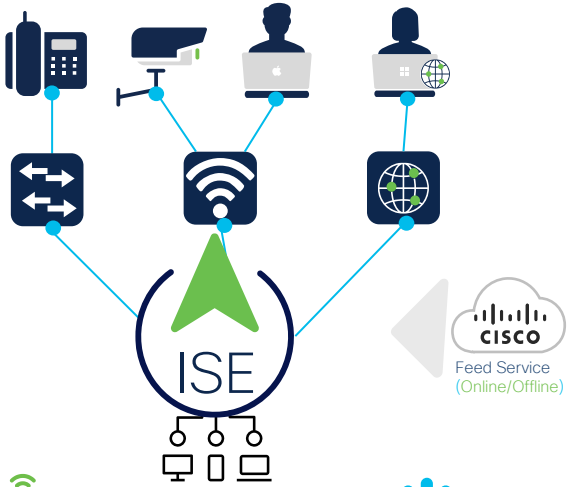
Context Reuse

by eco-system partners for analysis & control



Multi-Factor Classification on ISE

MFC-Manufacturer: Cisco
MFC-EndpointType: IP-Phone
MFC-Model: IP Phone 7980
MFC-OS: IOS



MFC-Manufacturer

MFC-EndpointType

MFC-Model

MFC-OS


Cisco


Arlo


Apple


Lenovo


IP-Phone


Camera


Laptop


Laptop


IP Phone 7980


MacBook Pro


Pro Wireless Cam


Thinkpad 540

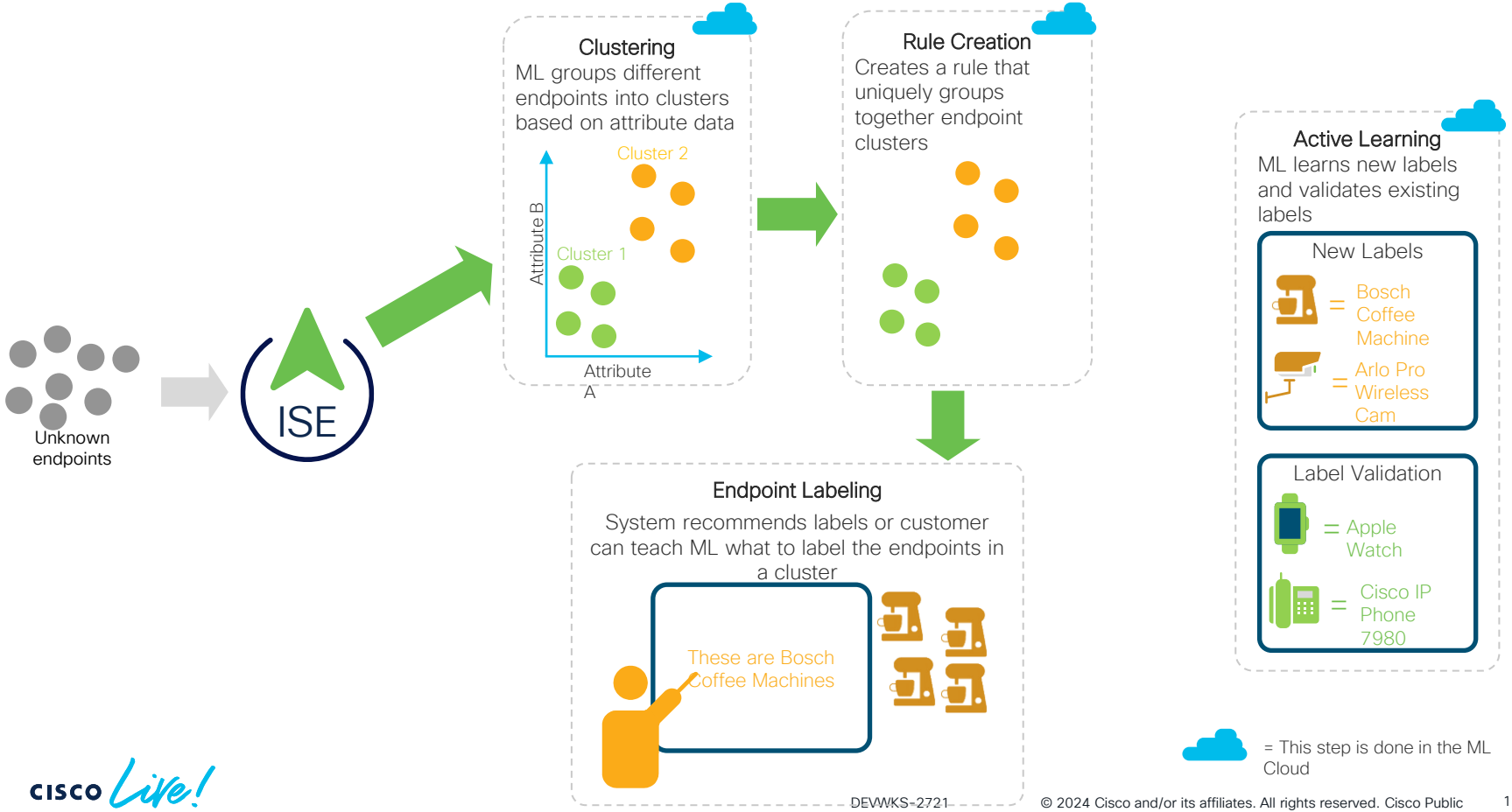

IOS


macOS 12.0.1

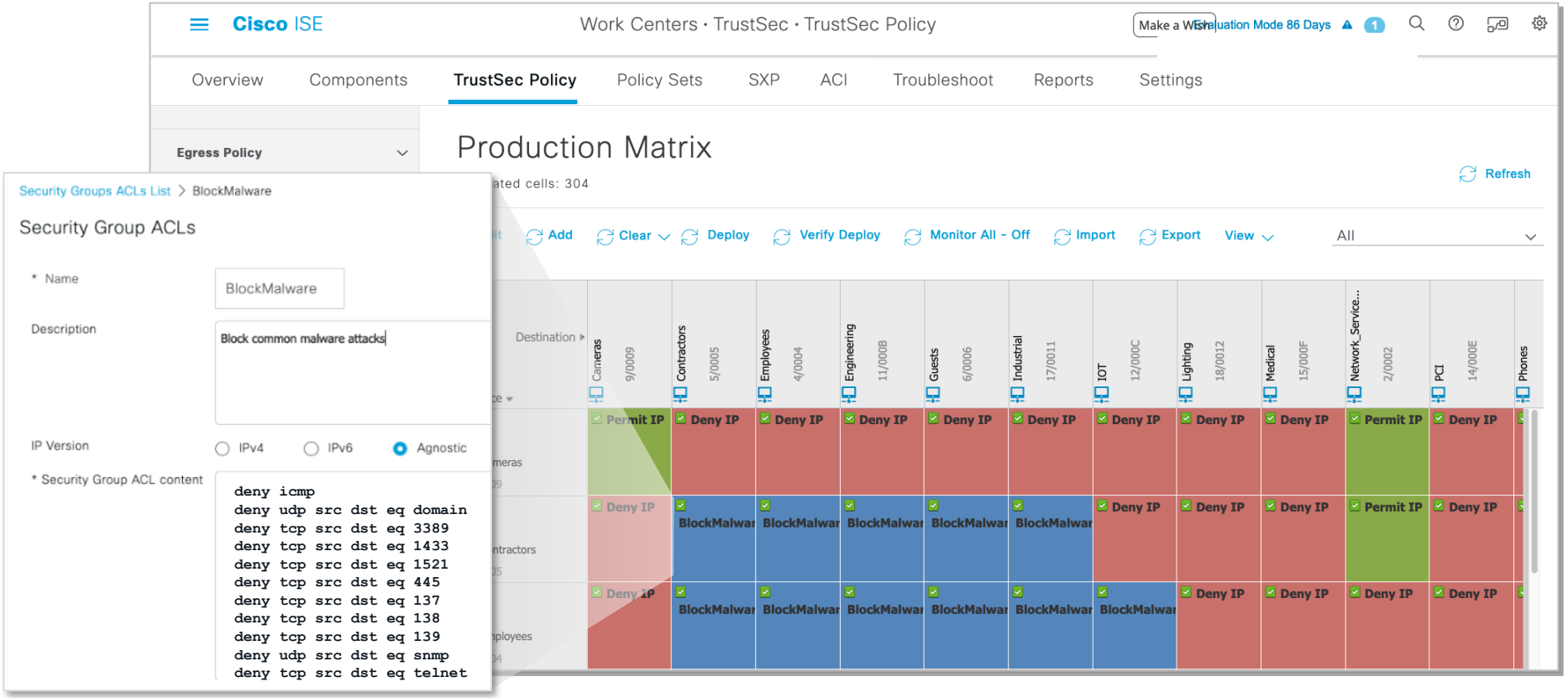

Linux


Windows Enterprise

Cisco AI Machine Learning Profiling

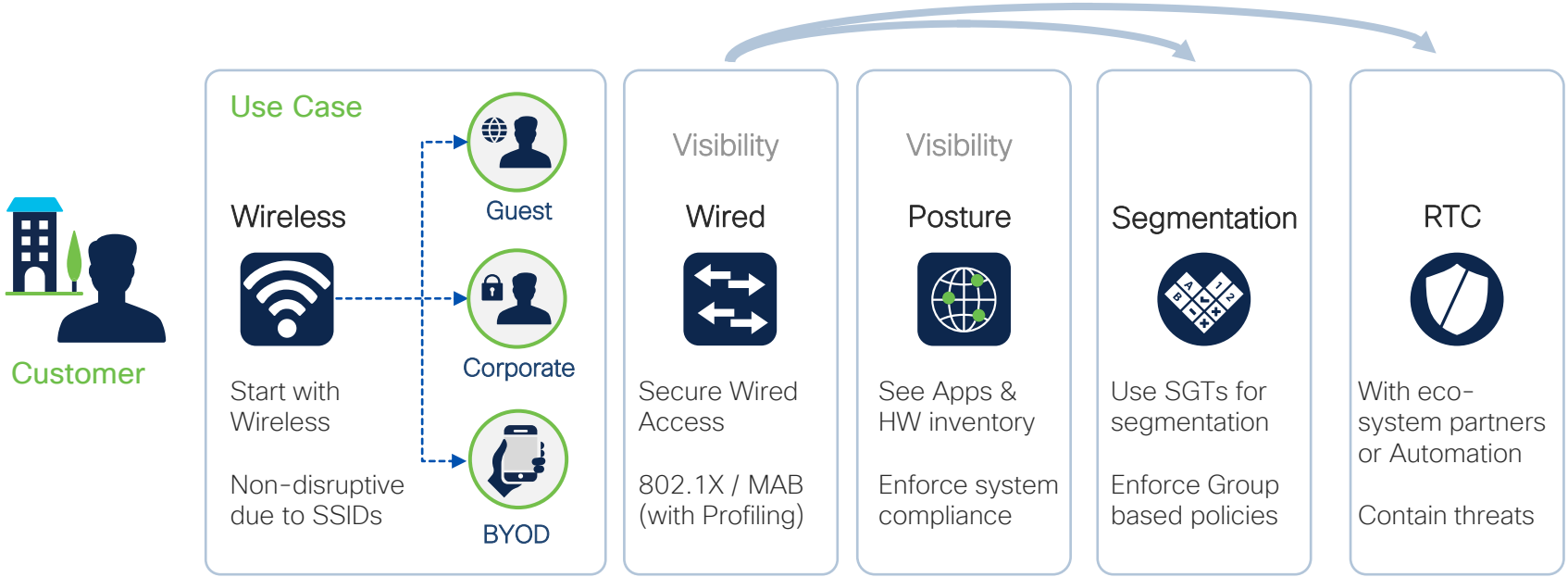


Non-Fabric Group-Based Policy Enforcement

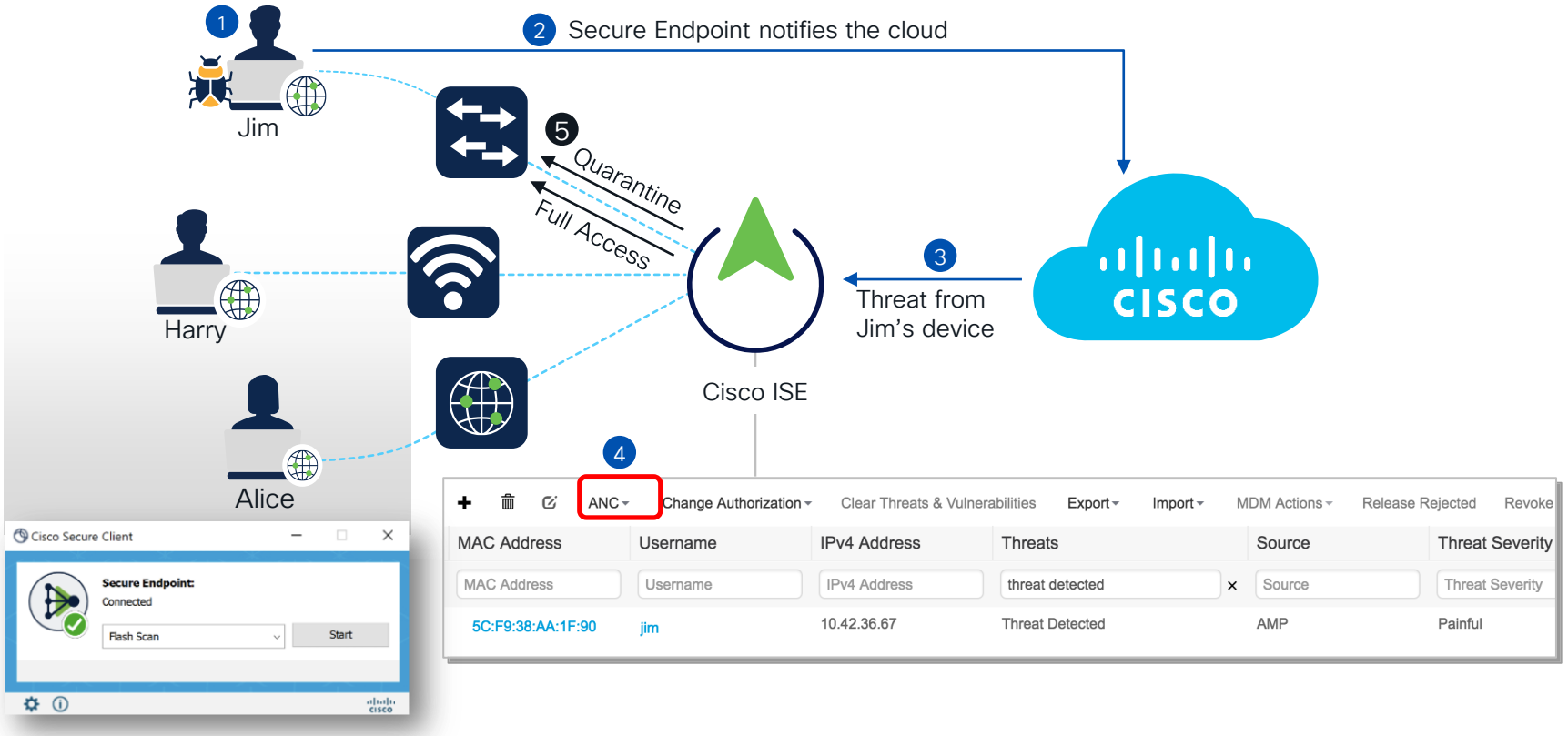


A Typical Customer Journey

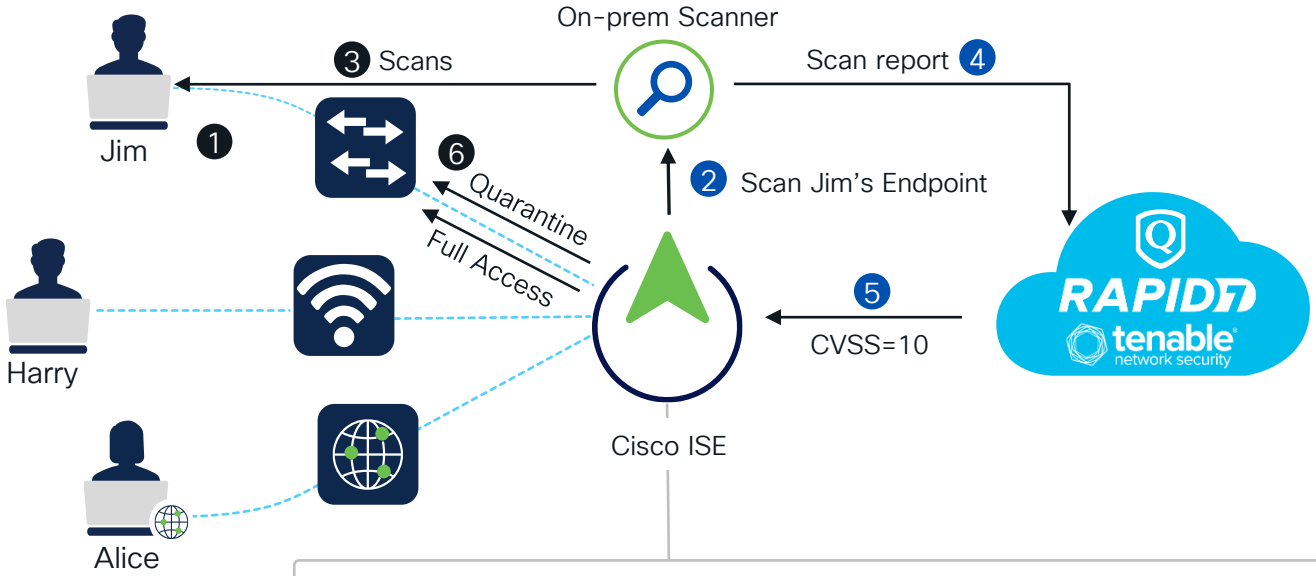
Not a standard or recommended approach
Each use case may be the end goal



Threat Visibility Rapid Threat Containment (RTC)



Vulnerability Assessment (Threat-Centric NAC)

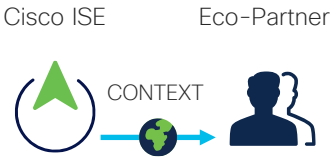


CVSS: Common Vulnerability Scoring System

Context Sharing with pxGrid

Eco system partnership to enrich, exchange context and enact

Context to Partner

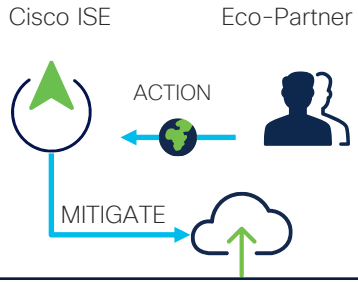


ISE makes Customer IT Platforms User/Identity, Device and Network Aware

Enrich ISE Context

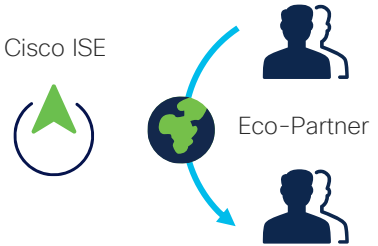
Enrich ISE context. Make ISE a better Policy Enforcement Platform

Threat Mitigation



Enforce dynamic policies into the network based on Partner's request

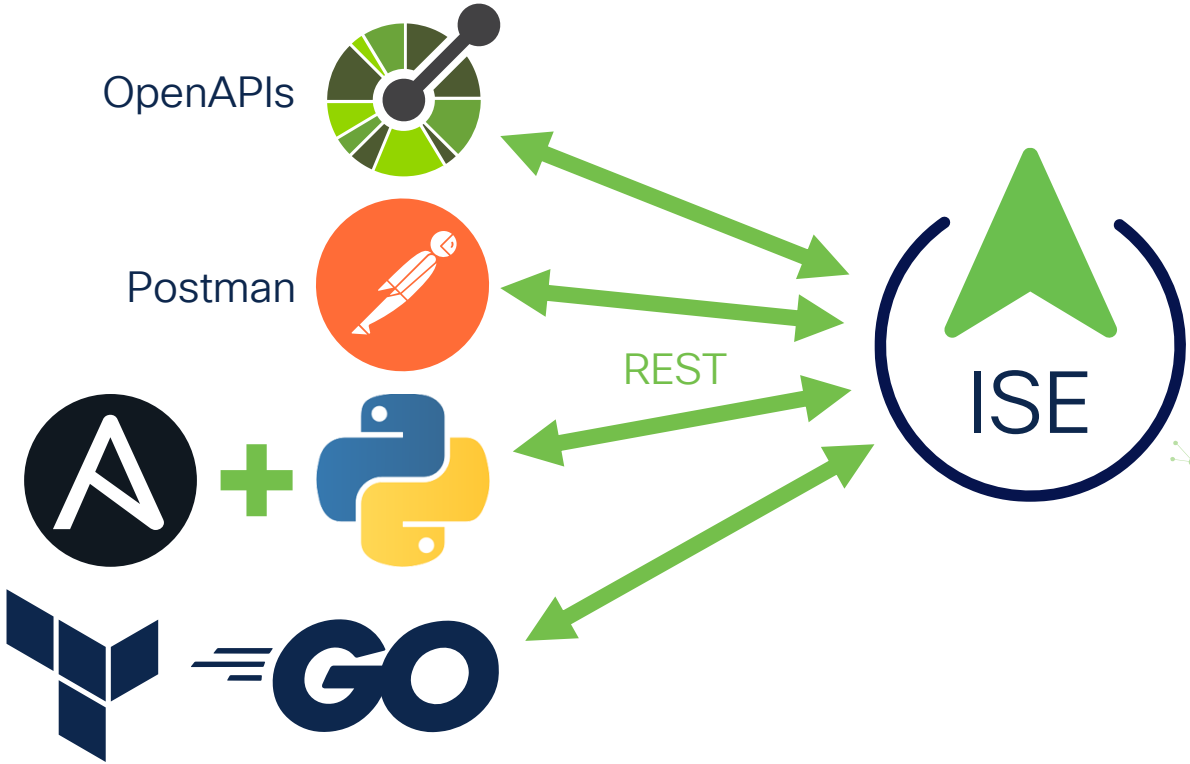
Context Brokerage

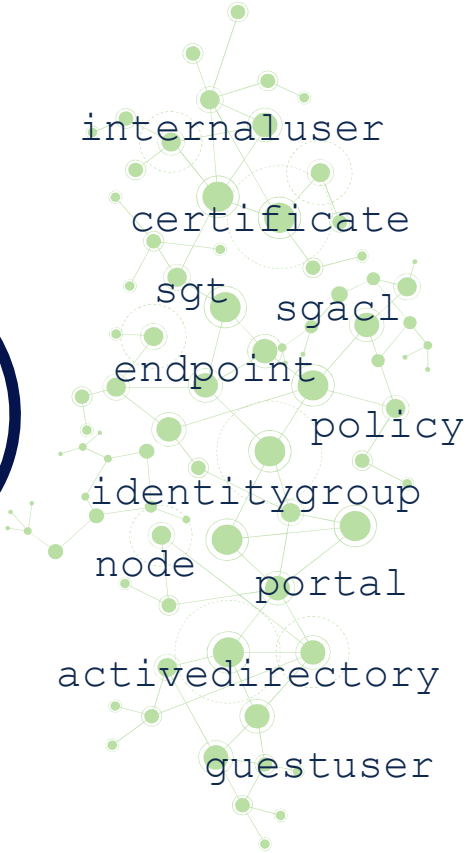


ISE 2.2+

ISE brokers Customer's IT platforms to share data amongst themselves

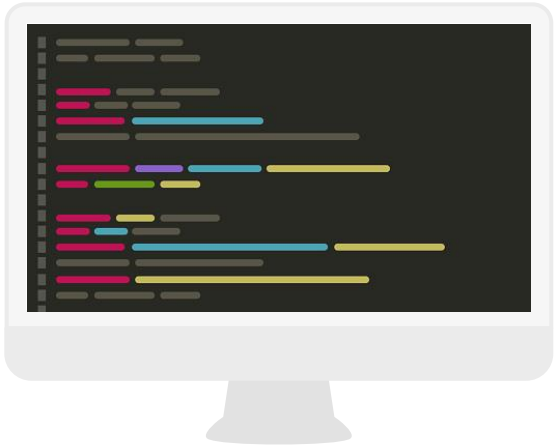
ISE APIs and Automation





github.com/CiscoISE

ISE Policy Management & Lifecycle Orchestration



ISE Zero-touch provisioning and Lifecyle management via APIs

ANSIBLE

TERRAFORM

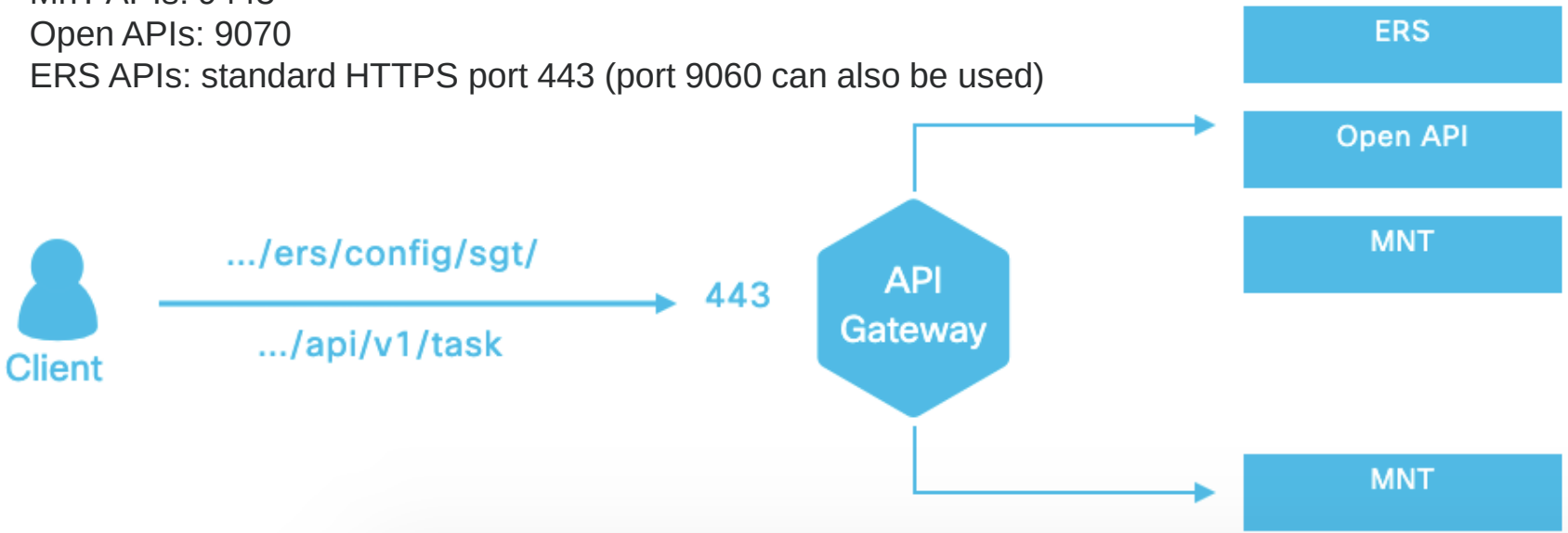
1. Zero Touch Provisioning
2. Patch/Hot-patch installation
3. Certificate Management
4. Deployment Automation
5. Policy Management
6. License Management
7. Configuration Backup and Restore

cs.co/ise-api

ISE API Gateway

From Cisco ISE Release 3.1 onwards, the MnT (Monitoring) APIs, the ERS APIs and the Open APIs all are routed through the API Gateway. The following ports need to opened between the API gateway node and all other nodes in the deployment for the respective APIs.

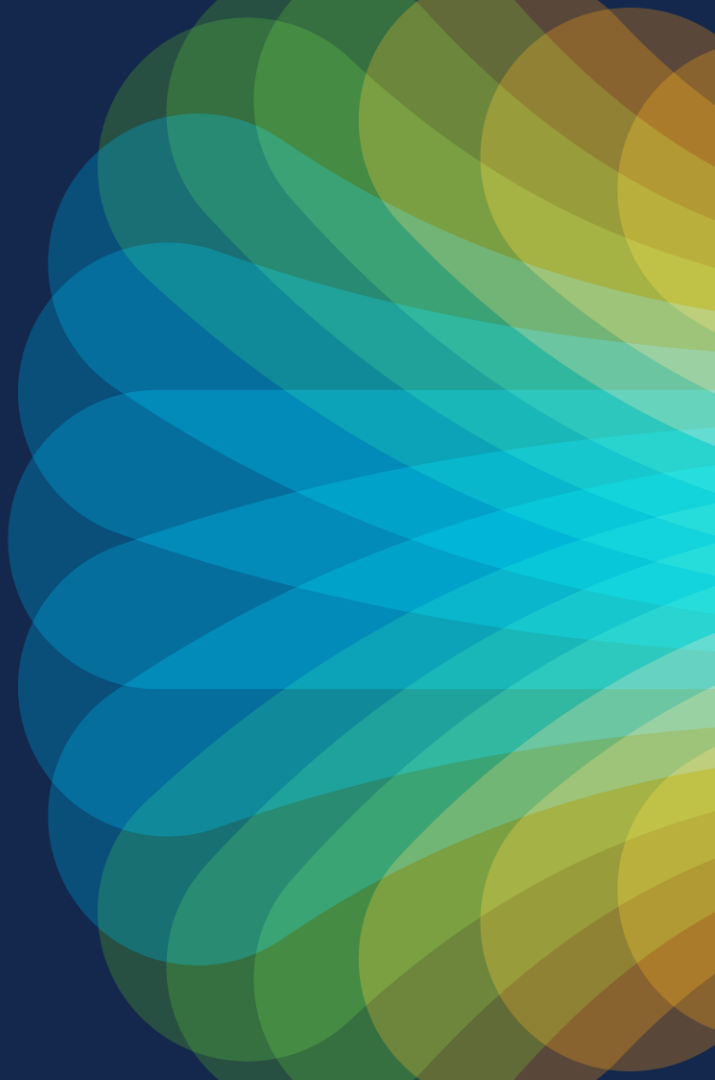
- MnT APIs: 9443
- Open APIs: 9070
- ERS APIs: standard HTTPS port 443 (port 9060 can also be used)



Lab!

Please access:

https://cs.co/ise



cisco DevNet

WE ARE CISCO DEVNET

Continue to Learn, Code, and Build with Cisco DevNet

Get access to exclusive resources including API documentation, Code Exchange, self-paced learning labs, sandboxes, community forums and events!

Scan QR Code to get started.

Answer to ISE List ANC policies

```
if __name__ == "__main__":
    #TODO : #1 Call the fucntion to get ISE ANC policies assign the value returned by function to "policies" variable
    policies = get_ise_anc_policies()
    print(
        white("\nAdaptive Network Control (ANC) Policies:", bold=True),
        pformat(policies),
        sep="\n"
    )
    #TODO: #2 Use the policy/polices you have received from ISE to get the details on the policy.
    devnet_anc_policy = get_ise_anc_policy_details("ANC_Devnet")
    #Or, the query can be get_ise_anc_policy_details(policies[0]['id'])

    #TODO: #3 Print the policy details you have received from ISE
    print(
        white("\nANC_Devnet Adaptive Network Control Policy:", bold=True),
        pformat(devnet_anc_policy),
        sep="\n"
    )
```

Answer to ISE optional Mission

```
#!/usr/bin/env python

import json, sys, os, requests
from pathlib import Path
from crayons import blue, green, red
from requests.packages.urllib3.exceptions import InsecureRequestWarning
from requests.auth import HTTPBasicAuth


# Locate the directory containing this file and the repository root.
# Temporarily add these directories to the system path so that we can import
# local files.
here = Path(__file__).parent.absolute()
repository_root = (here / ".." / "..").resolve()

sys.path.insert(0, str(repository_root))

# Disable insecure request warnings
requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

# Functions

def createPayload(maclist, policy):
    data_to_send = {
        'OperationAdditionalData': {
            'additionalData' : [{
                'name': 'macAddress',
                f'value': maclist
                },
                {
                    'name': 'policyName',
                    f'value': policy
                    }]
        }
    }
    return data_to_send


def readmacaddr_file(filename) :
    with open (filename, 'r') as fp:
        maclist = json.loads(fp.read())
    return maclist

headers = {
    'content-type': "application/json",
    'accept': "application/json"
    }
```

```
username = "admin"
password = os.getenv('SESSION_ID')
host = os.getenv('ISE_INTERNAL')
port = "443"
authentication = HTTPBasicAuth(username, password)

def get_policy_ise():

    #TODO: #1 Create the URL for the GET request to get the ANC policy from ISE. Hint: Make sure you pass the Auth paramenters for the API call
    url = f"https://{username}:{password}@{host}:{port}/ers/config/ancpolicy"

    #Create GET Request
    req = requests.get(url, verify=False, headers=headers, auth=authentication)
    #req = requests.request("GET", url, verify=False, headers=headers)
    namelist = " "
    if(req.status_code == 200):
        resp_json = req.json()
        policies = resp_json["SearchResult"]["resources"]
        for policy in policies:
            namelist = policy["name"]
            print("\nI've Found the Quarantine Policy {0} to Nuke the Rogue computers from the corp network... \n".format(namelist) )
    else:
        print("An error has ocurred with the following code %(error)s" % {'error': req.status_code})
    return namelist

def post_to_ise(maclist, namelist):
    #TODO: #2 Create the URL for the PUT request to apply the ANC policy! Hint: Make sure you pass the Auth paramenters for the API call
    url = f"https://{host}:{port}/ers/config/ancendpoint/apply"

    for items in maclist:
        payload = "{\r\n    \"OperationAdditionalData\": {\r\n    \"additionalData\": [{\r\n    \"name\": \"macAddress\",\r\n    \"value\": \""+ items + "\"\r\n    },\r\n    {\r\n    \"name\": \"policyName\",\r\n    \"value\": \"" + namelist + "" +
"\r\n    }]\r\n  }\r\n}"
        print(json.dumps(payload,sort_keys=True,indent=3))
        response = requests.request("PUT", url, data=payload, verify=False, headers=headers, auth=authentication)
        if(response.status_code == 204):
            print("Done!..Applied Quarantine policy to the rogue endpoint...MAC: {0} Threat is now contained....".format(items))
        else:
            print("An error has ocurred with the following code %(error)s" % {'error': response.status_code})

if __name__ == "__main__":
    maclist_path = repository_root / "mission-data/mac-addresses.json"
    maclist = readmacaddr_file(maclist_path)

    #TODO #3 Call the function for getting ANC policy and store it in the policylist variable
    policylist = get_policy_ise()

    #TODO #4 Call the function for applying policy to the endpoints
    post_to_ise(maclist, policylist)

    # # Finally, Display Mission Completed

    print(green("ISE Mission Completed!!!"))
```


cisco
The bridge to possible
Thank you
cisco Live!


cisco Live!
Let's go