Karim el-Zaben

Implementing Advanced Authentication and Access Control in Cisco SDA Network

Metropolia University of Applied Sciences

Bachelor of Engineering

Information and Communication Technology

Bachelor's Thesis

28.11.2025

Abstract

Author:	Karim el-Zaben
Title:	Implementing Advanced Authentication and Access Control in Cisco SDA Network
Number of Pages:	59 pages + 1 appendices
Date:	28 November 2025
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communication Technology
Supervisors:	Janne Salonen, Director of school

A large public sector organization is migrating to a software-defined networking architecture to achieve greater scalability, improved user mobility, simplified maintenance, intelligent monitoring, and granular access control through modern network automation methods. Prior to migrating from a traditional network architecture, the network environment must be thoroughly mapped, which requires accurately identifying and profiling all connected endpoint devices. The goal will be to deploy general and specialized methods for acquiring device attributes to support custom and pre-defined profiler policies, enabling more precise identification of previously unknown conventional and non-conventional endpoint devices. Since all tests and required configuration changes will be performed in a production environment, the chosen approach must be carefully researched and properly implemented to prevent any disruption to network performance or authorized access. Profiling tests were conducted in accordance with a tailored profiling plan, and the results provided sufficient insight to accurately identify the challenges associated with deploying error-free device probing in the target network environment. Before pursuing a broader probe deployment or more advanced probing methods, it would be strongly advised to address all issues revealed by the tests.

Keywords: Computer network, Network automation, Computer security, Software-defined networking, Network access control

---

The originality of this thesis has been checked using Turnitin Originality Check service.

Open AI’s ChatGPT has been used to support spelling, grammar, and language refinement.

Contents

List of Abbreviations

AAA	Authentication, authorization, and accounting
ACIDex	AnyConnect Identity Extensions
AD	Microsoft Active Directory
AI	Artificial Intelligence
API	Application programming interface
ARP	Address Resolution Protocol
AV	Attribute-Value
BACnet	Building Automation and Control networks
CBAR	Controller Based Application Recognition
CDP	Cisco Discovery Protocol
CF	Certainty Factor
CoA	Change of Authorization
CSV	Comma-separated values
DDI	Deployment Difficulty Index
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System

Dot1X	IEEE 802.1X
DPI	Deep Packet Inspection
EID	Endpoint Identifier
FQDN	Fully Qualified Domain Name
GBAC	Group-Based Access Control
GBP	Group-Based Policy
SIP	Session Initiation Protocol
HTDB	Host Tracking Database
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IND	Cisco Industrial Network Director
IoT	Internet of Things
IP	Internet Protocol
ISE	Cisco Identity Services Engine
L2	OSI model Layer 2
L3	OSI model Layer 3
L7	OSI model Layer 7

LISP	Location Identifier Separation Protocol
LLDP	Link Layer Discovery Protocol
MAB	MAC Authentication Bypass
MAC	Media Access Control address
MDM	Mobile Device Management
mDNS	Multicast DNS
NAC	Network Access Control
NAD	Network Access Device
NBAR	Network Based Application Recognition
NDG	Network Device Group
NII	Network Impact Index
NMAP	Network Mapper
ONF	Open Networking Foundation
OS	Operating System
OSI	Open Systems Interconnection
OUI	Organizationally Unique Identifier
PNAC	Port-based Network Access Control

PSN	Polilcy Service Node
PVI	Probe Value Index
pxGrid	Cisco Platform Exchange Grid
RADIUS	Remote Authentication Dial-In User Service
RLOC	Routing locator (LISP)
RBAC	Role-Based Access Control
SDA	Cisco Software Defined Access
SDN	Software-Defined Networking
SGT	Security Group Tag or Scalable Group Tag
SMB	Server Message Block
SNMP	Simple Network Management Protocol
SPAN	Switch Port Analyzer
TCF	Total Certainty Factor
TCP	Transmission Control Protocol
UDID	Unique Device Identifier
UDP	User Datagram Protocol
VACL	VLAN Access Control List

VLAN	Virtual Local Area Network
VN	Virtual Network
VNI	VXLAN Network Identifier
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding
VXLAN	Virtual Extensible LAN

1 Introduction

This bachelor's thesis is comissioned by a large public sector organization, Organization A, to research methods in automating the authentication and access control of conventional and non-conventional endpoint devices connected to the network, in order to prepare for a smooth and secure migration to a modern Software Defined Networking (SDN) architecture.

The configuration of Network Access Devices (NADs) and Cisco Identity Services Engine (ISE) will be executed by Organization A and its partner company (Partner B) personnel, respectively. Changes to these systems require senior supervision, expertise, and strict authorization.

Before work toward the project goal can begin, it is necessary to gain an understanding of the current network environment and general policies at Organization A, as well as the principles and best practices of an ideal Cisco Software-Defined Access (SDA) architecture. Reviewing the master's thesis Building Scalable LAN Solution [1], written by the Organization A supervisor, Mikko Nieminen, provides valuable insights into Organization A's perspectives on implementing an SDA solution. Mikko Nieminen's work offers a comprehensive current state analysis of Organization A's network architecture, technical issues and process challenges [1] and serves as an essential reference for achieving a thorough understanding of the existing network environment.

This thesis first presents the theoretical background necessary to understand the principles of Cisco SDA, followed by an examination of endpoint device identification, profiling, and access control concepts within the context of the intended architectural implementation.

After establishing a sufficient theoretical foundation, the study will proceed to apply the acquired knowledge. The network will be mapped, and appropriate endpoint identification and profiling methods will be selected for testing. The

results of the profiling tests will be analyzed and discussed in detail. Finally, conclusions will be presented, the overall outcomes of the project will be evaluated, and recommendations for future development and implementation will be provided.

2 Theoretical Background

2.1 SDN

The Open Networking Foundation (ONF) defines SDN as ”an emerging network architecture where network control is decoupled from forwarding and is directly programmable” [2]. Decoupling and centralizing of control and data planes gives network management new possibilities in defining network access, security and performance by programmatically automating tasks that would have traditionally been tedious manual labour with risk of misconfiguration or accumulation of undesirable complexity. [2]

The transition from a traditional network to a SDN architecture presents significant challenges, particularly around device interoperability and the migration of endpoint authentication and access-control mechanisms to a SDN-driven model. Therefore, the implementation of SDN will most likely be evaluated within a limited segment of the larger network environment and reviewed for security, accessibility and performance concerns before any further expansions take place. [2]

2.1.1 SDN Reference Model

The ONF defines a reference model for SDN that organizes network functions into three distinct, basic components: the application layer, the control layer, and the infrastructure layer. The component stack has been illustrated in Figure 1.

At the base of the model, the data plane represents the infrastructure layer and is composed of forwarding devices such as switches and routers. These devices perform two key functions: collecting and reporting network status information to controllers, and processing packets according to forwarding rules dictated by the controller plane. [2][3]

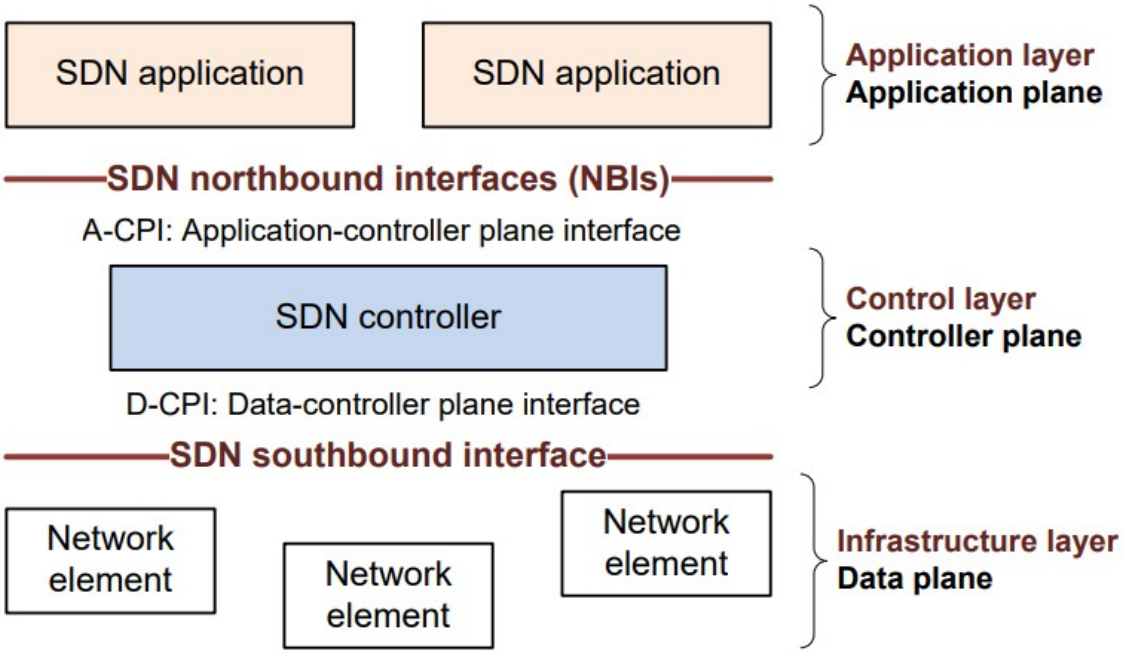


Figure 1: SDN refrence model with its basic components. [3]

The control layer serves as the intermediary between applications and the underlying infrastructure as the controller plane. Through its southbound interface, it communicates with network devices by installing forwarding rules and retrieving network state information. Through its northbound interface, it exposes service access points, typically in the form of Application Programming Interfaces (APIs), enabling applications to query network conditions and enforce policies. In large-scale deployments, multiple controller nodes are often required, which necessitates east–west communication for coordination and synchronization among the nodes. [2][3]

At the top, the application layer encompasses SDN applications designed to fulfill user and organizational requirements as the application plane. These applications leverage the programmable platform provided by the control layer to influence forwarding behavior and optimize network operation. Common examples include dynamic access control, seamless mobility, server load balancing, and network virtualization. [2][3]

Overall, the ONF's SDN reference model illustrates how the logical separation of infrastructure, control, and application layers introduces programmability, flexibility, and centralized management into modern networks. This framework provides a foundation upon which vendors can develop their own SDN-based solutions.

2.2 Cisco SDA

SDA is Cisco's intent-based networking solution for enterprise campus environments. It applies the concepts of SDN, like the seperation of control and data layers, network virtualization, and centralized management. Cisco SDA divides its architecture into four operational planes: data, control, management and policy. The operational planes are facilitated through fabric roles (Figure 2), each responsible for a distinct set of functions. [4]

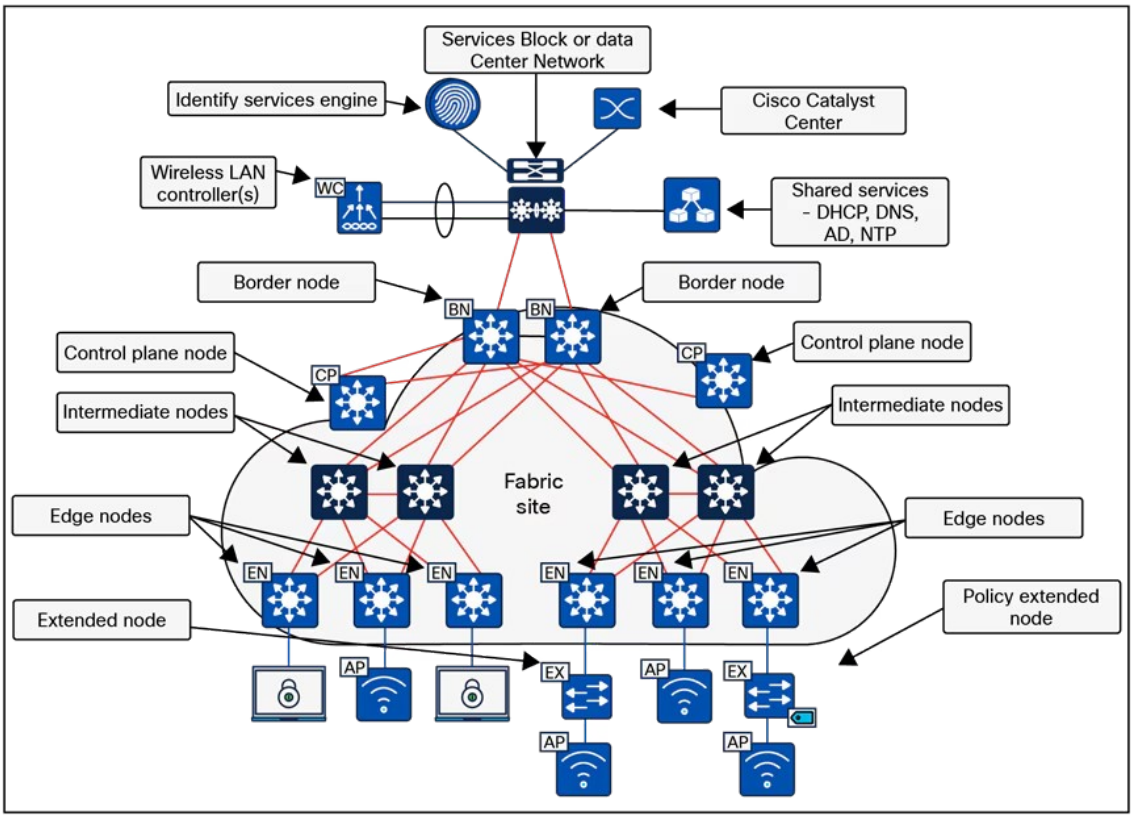


Figure 2: SDA fabric roles. [4]

SDA separates the network into two complementary layers as illustrated in Figure 3: the underlay and the overlay. The underlay is the physical network infrastructure, responsible for providing basic connectivity between network devices. In contrast, the overlay is a virtualized layer that abstracts the physical network and enables a logical, programmable connection between endpoints. Together they form, what Cisco calls, the Fabric. [4]

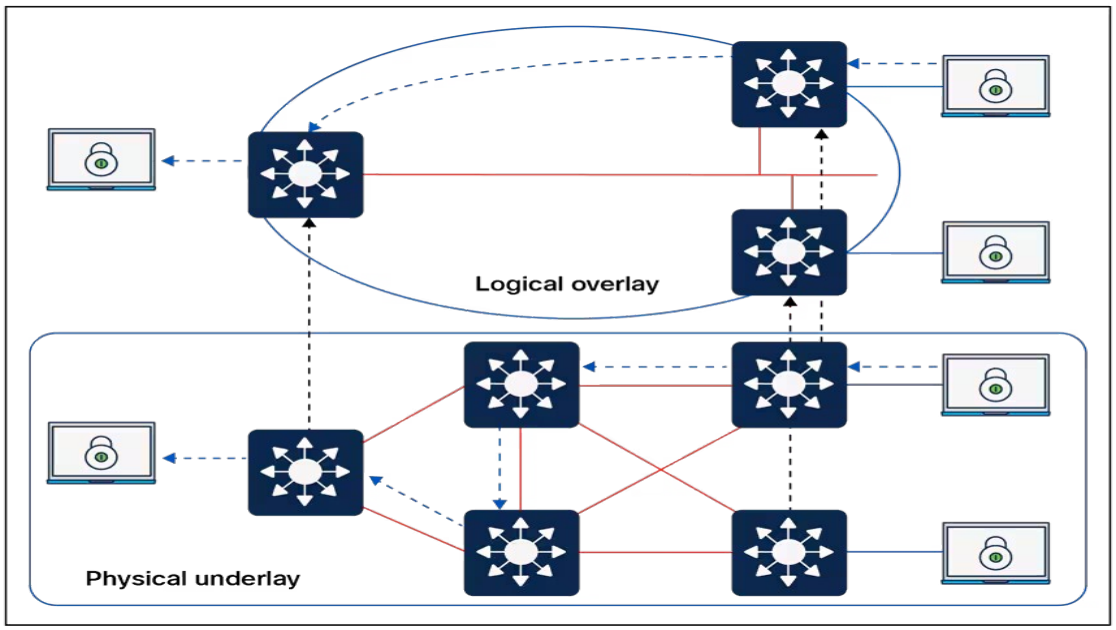


Figure 3: The overlay on top of the physical underlay. [4]

A comprehensive and in depth description of SDA is beyond the scope of this thesis, which focuses on the goal of profiling unknown endpoint devices. The following sections give a brief overview of each operational plane used in SDA.

2.2.1 Data Plane

A core principle of SDA is the Overlay, enabled by the Data Plane's Virtual Extensible LAN (VXLAN), a MAC-in-IP encapsulation method that utilizes User Datagram Protocol (UDP) for transporting OSI Layer 2 (L2) frames, across OSI Layer 3 (L3) infrastructure. Each overlay network is identified by a VXLAN Network Identifier (VNI) in the VXLAN header, which can be leveraged for policy-based forwarding decisions. [4]

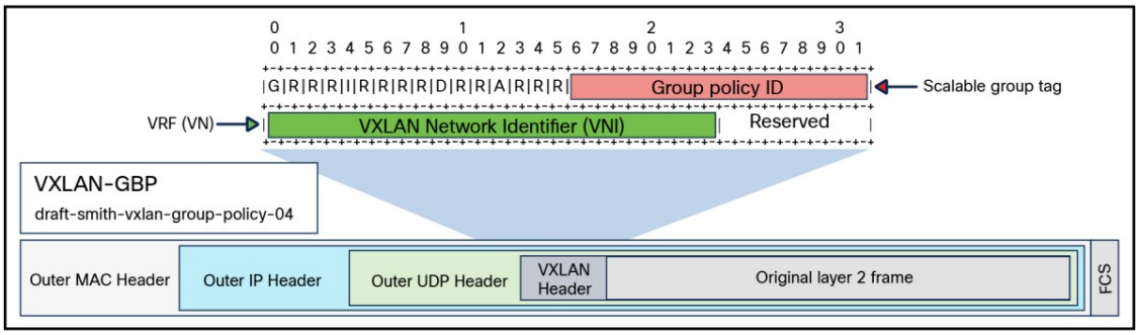


Figure 4: VXLAN-GBP header [4]

VXLAN supports up to 16,777,216 overlay segments via its 24-bit VXLAN Network Identifier (VNI). Cisco's VXLAN-GBP extension repurposes 16 previously reserved bits in the VXLAN header to carry Security Group Tags (SGTs), enabling up to 65,536 distinct SGT values for granular identity based segmentation. [4]

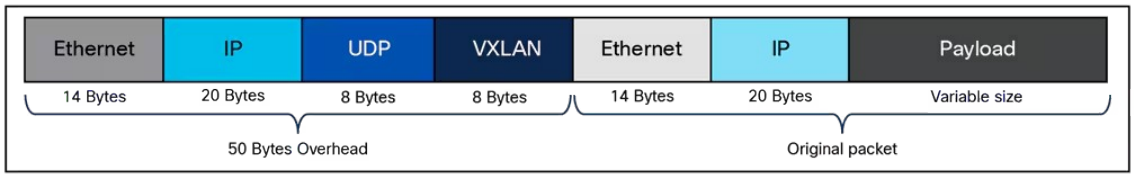


Figure 5: Encapsulation MTU overhead.

The encapsulation method creates additional Maximum Transmission Unit (MTU) overhead adding an extra 50 bytes to the original packet, as seen in detail in Figure 5. [4]

2.2.2 Control Plane

The Control Plane is driven by the Location Identifier Separation Protocol (LISP), which separates an endpoint device's identity from its location by performing an EID-to-RLOC mapping. LISP treats an endpoint's IP address as its Endpoint Identifier (EID), so the IP address stays the same even when the client moves between different attachment points, only changing it's Routing Locator (RLOC). [4][5]

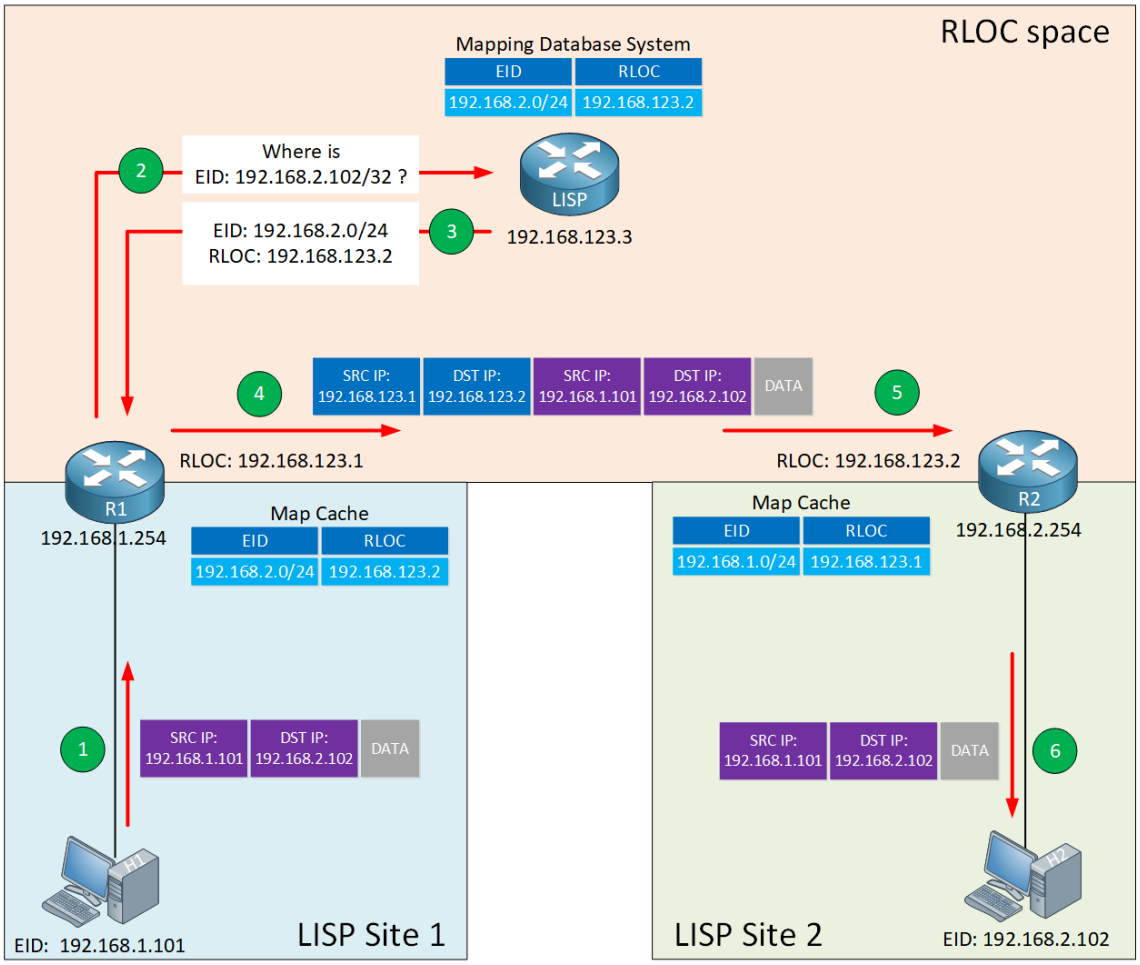


Figure 6: High-level simplified overview of how LISP works. [6]

During endpoint discovery a control plane node, acting as a Map-Server and a Map-Resolver, registers the EID and maps it against a Routing Locator (RLOC) that points to the location of the fabric edge node hosting the endpoint. By registering an EID-to-RLOC entry to the control plane node's Host Tracking Database (HTDB), an edge node can query the control plane nodes to resolve the correct EID-to-RLOC mapping, allowing the edge to forward traffic to the correct site. If the RLOC cannot be resolved within the fabric, traffic is forwarded to the default border node for further processing to achieve inter-fabric connectivity through SDA transits. The control plane function is typically handled by two to six dedicated control plane nodes depending on the size of the network, providing redundancy and load balancing. [4][5]

2.2.3 Management Plane

Management of SDA is centralized in Cisco Catalyst Center (formerly known as Cisco DNA Center), automating device onboarding and configuration, and translating user's intent programmatically into device-level changes to enforce policy together with ISE across wired and wireless fabric devices. Catalyst Center ensures consistent and scalable deployment of the fabric's control, data, and policy planes, while providing comprehensive visibility of the network infrastructure. [4][7]

Catalyst Center can be integrated with advanced Artificial Intelligence (AI) and machine learning capabilities to proactively monitor network health, predict potential failures, and accelerate issue resolution. [7]

2.2.4 Policy Plane

The policy plane is implemented through Cisco TrustSec (CTS) centrally managed by ISE, an ”identity-based network access control and policy enforcement system” [9]. ISE together with Cisco Catalyst Center provides a comprehensive Network Access Control (NAC) framework. [4]

CTS's Role-Based Access Control (RBAC) and Group-Based Access Control (GBAC), provide secure network access control by dynamically mapping users and devices into profiles and logical-profile groups for consistent, identity-based policy enforcement across the network. Policies can be dynamically enforced using SGTs for identity-based microsegmentation across the fabric, while VNIs provide network-level segmentation. [4]

Configuring our target ISE Policy Service Nodes (PSNs) will be required in implementing advanced authentication and access control using SDA.

2.3 Endpoint Device Identification and Access Control in SDA

A more dynamic approach to access control is needed, as network environments become increasingly saturated with diverse endpoint devices subjected to novel requirements driven by innovations in technology and workplace culture. The traditional perimeter-based model of securing networks is insufficient in an era defined by mobility, cloud adoption, and the proliferation of IoT devices.

Authenticating clients are divided into supplicant and non-supplicant devices. IEEE 802.1X (dot1x) is a Port-Based Network Access Control (PNAC) protocol in which a supplicant device initiates an Extensible Authentication Protocol (EAP) exchange with the edge node acting as the authenticator. The authenticator forwards those EAP messages to an Remote Authentication Dial-In User Service (RADIUS) server for credential validation. On a successful authentication, the authenticator will enforce the authorization result. Non-supplicant devices that cannot perform this exchange rely instead on manual or fallback methods like MAC Authentication Bypass (MAB). Authorization dictates what an authenticated user or device is permitted to access and what actions they may perform.

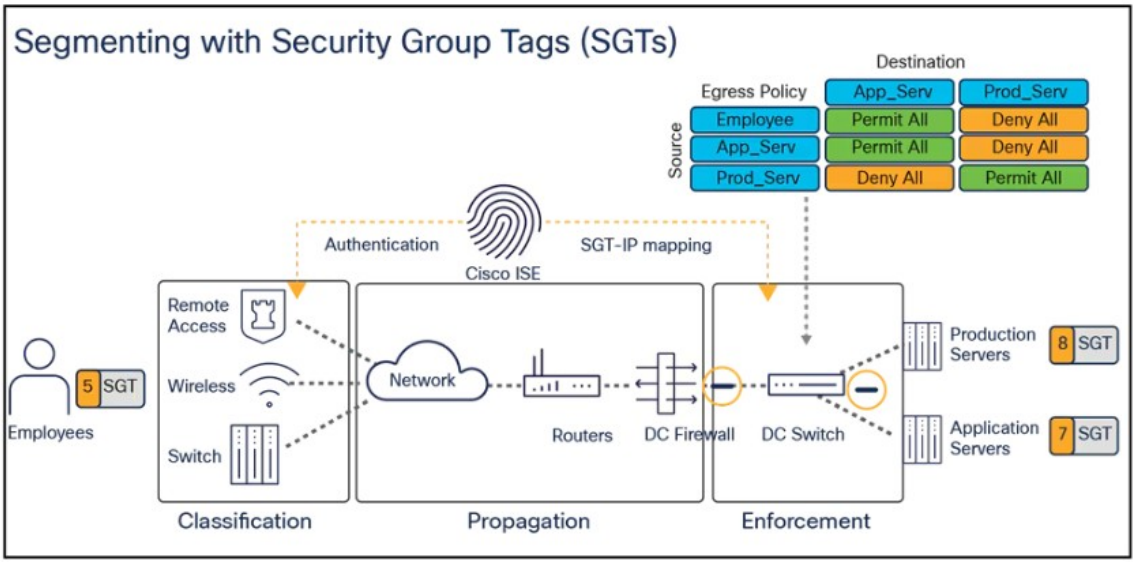


Figure 7: Cisco ISE segmentation use case using SGTs. [8]

For further control, SDA provides mechanisms to segment network access on a macro and micro level. Macrosegmentation is used for the isolation of network traffic into Virtual Networks (VNs). The VNI tag inside of the 24-bit VXLAN header is used in identifying the VN assigned to the endpoint. The L3 VNI maps to a Virtual Routing and Forwarding (VRF) instance and the L2 VNI maps to a Virtual Local Area Network (VLAN) broadcast domain. Microsegmentation is the isolation of endpoints inside of a VN into device- or user-type-based seucity groups. For example, a fisht tank sensor will be isolated to a VN called ”Casino IoT” and inside of that VN it will be assigned to a profile or a logical group of profiles called ”Aquaria Devices” with it’s own device group specific access control. The benefit of microsegmentation using SGTs comes in the ability to give fine-grained control over the access of the virtual network based on the identity of the user or device. Figure 7 illustrates a high level overview of how SGTs are used in group-to-group communication. [8][9]

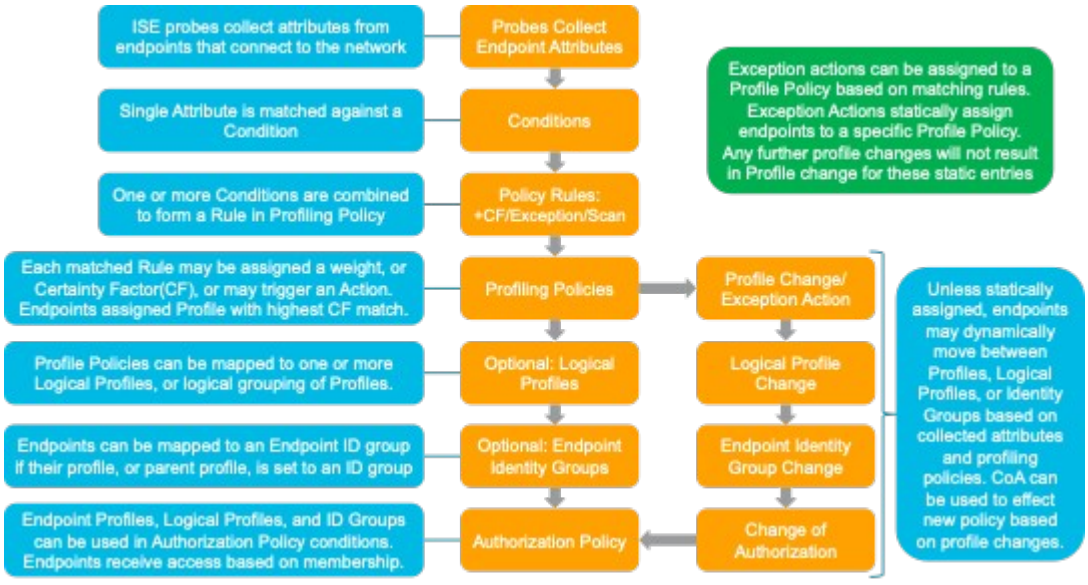


Figure 8: ISE Profiling Service process flow. [9]

The authentication of an endpoint device should not rely solely on correct user login credentials. It should also verify that the endpoint is a known and trusted device operating within a secure network environment. A previously unknown endpoint device is typically identified and assigned to a specific device profile or logical profile group. It is then granted authorization based on the policies

associated with that profile, after which its activity continues to be monitored to ensure compliance and security. Figure 8 shows a high level view of the ISE Profiling Service process flow, which we will be going through in the next sections. [9]

2.3.1 Gathering Profiling Attributes using Probes

Accurate endpoint identification requires the collection of profiling attributes from each endpoint device through the deployment of probes, which are configured and managed within ISE. Probes are specific for the data they collect and have varying impact on networking overhead when deployed. It is advisable to create a profiling plan for each NAD based on the types of endpoint devices it hosts, to ensure that only the probes necessary for successful profiling are enabled. [9]

The most efficient and Cisco-recommended probe is the device sensor. The device sensor collects CDP, LLDP and DHCP attributes from wired endpoints natively on the NAD and forwards them to ISE PSNs as Attribute-Value (AV) pairs inside RADIUS accounting packets. A maximum of 32 endpoints can be monitored per port, and an inactivity timer ages out sessions after 12 hours. [9]

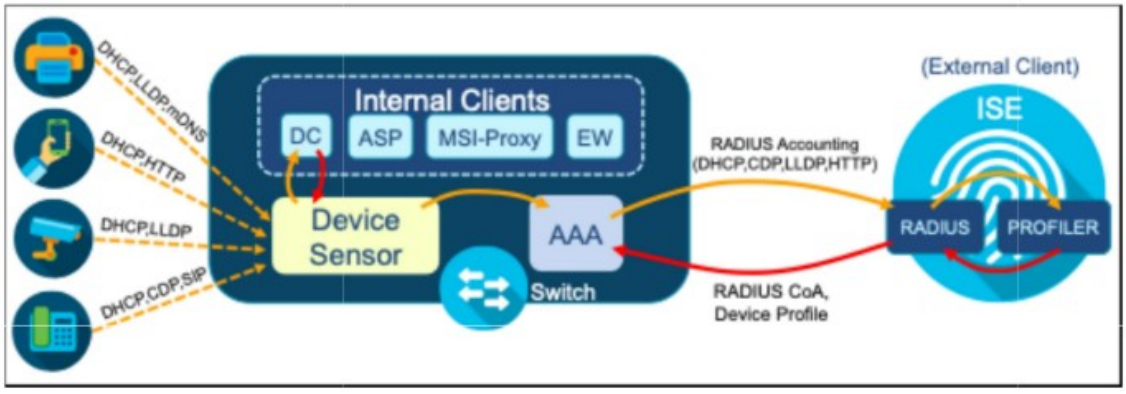


Figure 9: High level view of device sensor operation illustrated. [9]

Monitor Mode allows a network in the discovery phase of an SDA migration to observe authentication and authorization events without enforcing access

policies. In contrast, Closed Mode may block authentication attempts from legitimate users and devices that are not yet compatible with dot1x, hindering the collection of valuable endpoint attribute data. During the discovery phase, it is recommended to disable Change of Authorization (CoA) to prevent frequent updates to an endpoint device's authorization when initial profiling changes occur. [9]

In addition to the device sensor, ISE supports a wide range of probes, designed to collect specific profiling attributes (Table 1). Deployment should be in accordance with the defined profiling plan, ensuring both comprehensive visibility and controlled operational impact.

Table 1: Summary of probes, their key profiling attributes and use cases. [9]

Probe	**Key Profiling Attributes**	**Common Endpoint Profiling Use Cases**
RADIUS	MAC Address (OUI) IP Address NDG values	MAC Address > OUI = Indication of device vendor. Some endpoints can be profiled with this attribute alone if vendor only makes specific devices. Ex: Third-party IP phones, mobile devices, game consoles; MAC-to-IP bindings and probe support. NDG values for Location and Device Type may be used to classify based on connected NAD.
RADIUS w/Device Sensor	CDP/LLDP DHCP User-Agent mDNS H323/SIP	See SNMP probe for CDP/LLDP info. See DHCP probe for DHCP info. See HTTP probe for User-Agent info. mDNS, H323, and SIP offer unique insight into endpoint type and applications.
RADIUS w/ACIDex	MAC Address UDID Operating System Platform/Device Type	Extremely useful for profiling remote access VPN clients including client UDID for correlating to ISE and MDM endpoints. See RADIUS probe for MAC info. Detailed OS, platform, and device type

		information as visible to local AnyConnect agent.
SNMP	MAC Address/OUI CDP/LLDP ARP tables	See RADIUS probe for MAC info. Valuable for any vendor that uses CDP/LLDP. For example, Cisco IP phones, cameras, access points, appliances. Polling of device ARP tables populates ISE MAC-to-IP bindings.
DHCP	DHCP	Unique vendor IDs for hardware and software. DHCP fingerprints for OS detection. Hostname/FQDN for common name patterns may indicate OS or device type. Customer-defined identifiers. Additionally, provides MAC-to-IP bindings to support other probes.
DNS	FQDN	Value will depend on whether common naming conventions used for hostname/DNS.
HTTP	User-Agent	Operating system detection; some browsers like Chrome may mask actual OS.
NetFlow	Protocol Source/Dest IP Source/Dest/Ports	Good for detecting mission-specific endpoints with unique traffic patterns or use general purpose hardware/software. May detect anomalous traffic for specific endpoints.
NMAP	Operating System Common and custom ports Service Version Info SMB data Endpoint SNMP data	Operating system detection IF scanning not blocked by network/client FW. Good for detecting endpoints that listen on well-known or specific UDP/TCP ports. Detect endpoints by running services, applications, and their version. Useful for Windows clients to collect domain and OS version data. Offers classification of endpoints that run SNMP agents like network printers and cameras.

AD	Exists in AD Operating System and Version AD Domain	Verification of managed AD hosts, domain, and OS details as known to Active Directory.
pxGrid	IoT Asset Custom Attributes	IoT asset attributes learned from reliable sources able to collect data typically unavailable to other probes. See Custom Attributes for additional info.
<Custom Attributes>	Customer defined	Highly flexible custom classifications for virtually any attribute associated to endpoints—classification, role, compliance, threat, asset data, ownership, etc.

Tables 3 and 4 present the rating of Deployment Difficulty Index (DDI), Network Impact Index (NII), and Probe Value Index (PVI) for each probe on a 1 to 3 scale (definitions in Table 2). Table 3 shows ratings for deployment during a discovery phase of wired devices, and Table 4 shows ratings for deployment on a RADIUS-authenticated wired network.

Table 2: Legend for the metrics and ratings used in Table 3 and Table 4. [9]

Metric		**Rating**		
Name	**Description**	**1**	**2**	**3**
DDI	Deployment Difficulty Index	Easy	Medium	Difficult
NII	Network Impact Index	Low Impact	Medium Impact	High Impact
PVI	Probe Value Index	High Value	Medium Value	Low Value

Table 3: Recommended best practices and guidance for probe selection during the discovery phase of wired endpoint devices. [9]

Probe (Method)	**DDI**	**NII**	**PVI**	**Key Profiling Attributes**	**Notes**
RADIUS	-	-	-	N/A	Not applicable since ISE not in auth control plane.
RADIUS w/ Device Sensor	2	2	1	CDP/LLDP DHCP User Agent mDNS/H323/SIP	If network supports Device Sensor, you can use RADIUS accounting independent of auth control plane. Network impact generally low, but should also monitor NAD impact.
RADIUS w/ACIDex	-	-	-	N/A	Not applicable since ISE not in auth control plane.
SNMPTrap	1	1	1	LinkUp/Down Traps MAC Notify Traps Informs	Detect endpoints connections / trigger SNMPQuery probe.
SNMPQuery	1	2	1	MAC Address (OUI) CDP/LLDP ARP tables	Polling of device ARP tables populates ISE MA-to-IP bindings. Be careful of high SNMP Query traffic triggered by excessive RADIUS accounting updates due to reauth or interim updates.
DHCP (Helper)	2	2	1	DHCP	Provides MAC-to-IP bindings. Network impact generally low, but be careful of low DHCP lease timers, re-DHCP due to network transitions, and DHCP Inform activity.
DHCP	2	3	1	DHCP	Provides MAC-to-IP

SPAN					bindings
DNS	1	1	2	FQDN	Value will depend on whether common naming conventions are used.
HTTP (Redirect)	-	-	-	N/A	Not applicable since ISE not in auth control plane.
HTTP (SPAN)	2	3	1	User-Agent	Consider SPAN of key HTTP chokepoints like server or Internet edge using intelligent SPAN/tap solutions and/or VACL Capture.
NetFLow	3	3	2	Protocol Source/Dest IP Source/Dest Ports	Recommended only for specific use cases, not general profiling.
NMAP	1	2	2	Operating System Open TCP/UDP ports w/Service Version Windows SMB Endpoint SNMP data	SNMP data assumes UDP/161 open and public string. Relative value of NMAP will depend on customer network and client configuration. OS results can be unreliable, but unique SMB and unique ports can help fill detection gaps.
AD	1	1	1	AD Asset Operating System	Endpoint AD membership and OS details. Require ISE AD join. Contingent on acquiring host/machine name from DHCP or DNS. AD data accessible even if not used for ISE authentication.
pxGrid	2	1	1	IoT Asset Attributes Custom Attributes	Requires external source that is pxGrid publisher. Custom attributes sent are determined by source, not

					ISE. Recommended for IoT devices and other sources of unique endpoint context.
Custom Attributes	2	1	1	Custom Attributes	Requires population of endpoint attributes via import, API, pxGrid, or manual entry. Useful for IoT or any endpoint where user-defined attributes can be extracted from external source and contribute to classification, trust, and state.

Table 4: Recommended best practices and guidance for probe deployed in a wired network configured for RADIUS authentication. [9]

Probe (Method)	**DDI**	**NII**	**PVI**	**Key Profiling Attributes**	**Notes**
RADIUS	1	2	1	N/A	Fundamental probe for device detection and enabling other probes.
RADIUS w/ Device Sensor	2	2	1	CDP/LLDP DHCP mDNS/H323/SIP	If running Cisco Catalyst Switches with Device Sensor support, then this is ideal and optimized method to collect select attributes.
RADIUS w/ACIDex	-	-	-	N/A	Not currently applicable to wired networks.
SNMPTrap	1	2	3	LinkUp/Down Traps MAC Notify Traps Informs	Detect endpoints connections / trigger SNMP Query probe. Not required with RADIUS AAA.

SNMPQuery	1	2	1	MAC Address (OUI) CDP/LLDP ARP tables	Polling of device ARP tables populates ISE MAC-to-IP bindings; Be careful of high SNMP Query traffic triggered by excessive RADIUS Accounting updates due to re-auth or Interim Updates.
DHCP (Helper)	2	2	1	DHCP Attributes	Provides MAC-to-IP Bindings; Be wary of low DHCP lease timers.
DHCP SPAN	2	3	1	DHCP Attributes	Provides MAC-to-IP Bindings
DNS	1	1	2	FQDN	Value will depend on whether common naming conventions used
HTTP (Redirect)	2	1	1	User-Agent	Value will depend on relative importance of OS for wired access.
HTTP (SPAN)	2	3	1	User-Agent	Consider SPAN of key HTTP chokepoints like Internet edge; Leverage smart SPAN solutions and VACL Capture if possible
NetFLow	3	3	2	Protocol Source/Dest IP Source/Dest Ports	Recommended only for specific use cases, not general profiling.
NMAP	1	2	2	Operating System Open TCP/UDP ports w/Service Version Windows SMB Endpoint SNMP data	SNMP data assumes UDP/161 open and public string. Relative value of NMAP will depend on customer network and client configuration. OS results can be unreliable, but unique SMB and unique ports can help fill detection gaps.

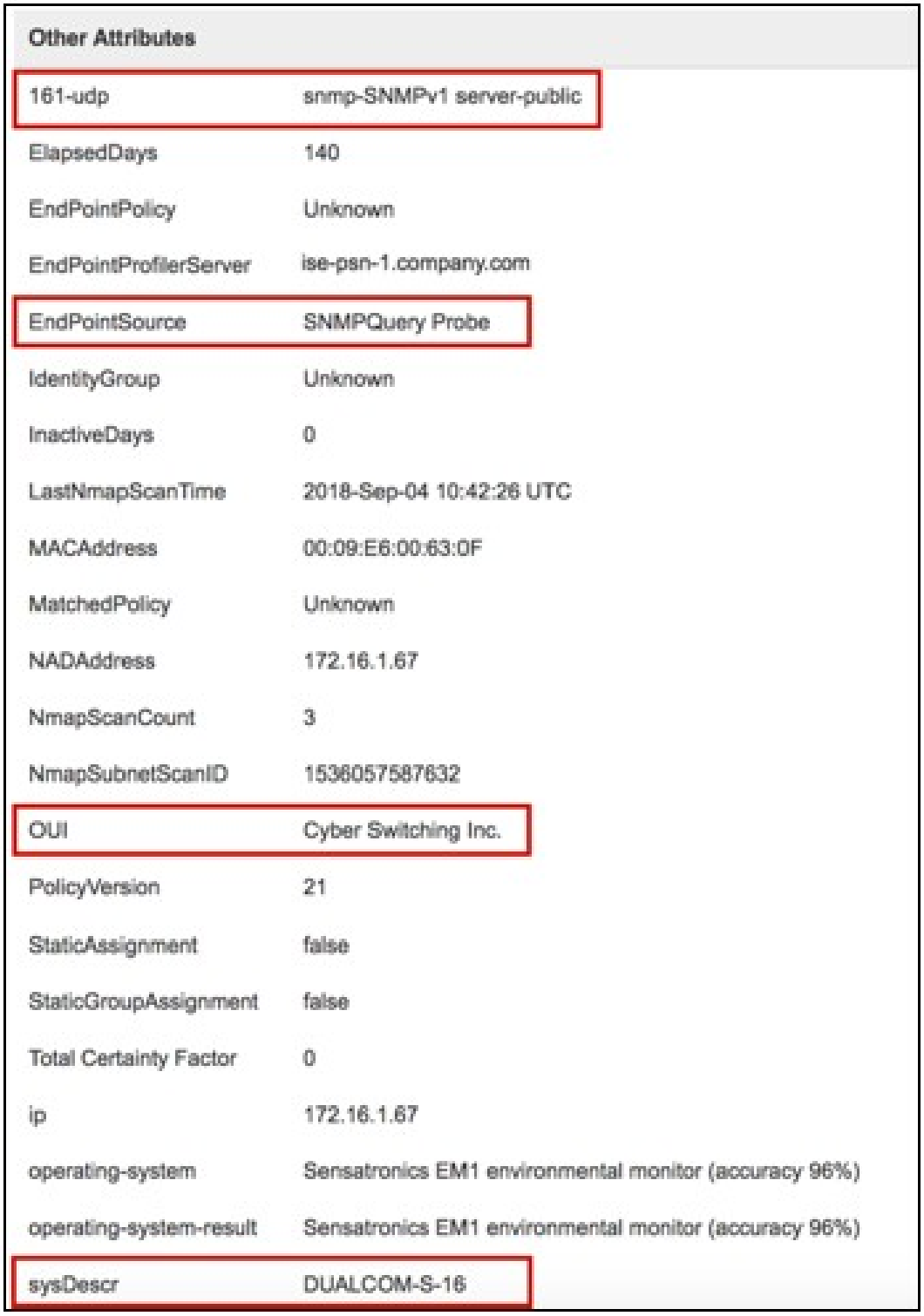
AD	1	1	1	AD Asset Operating System	Endpoint AD membership and OS details. Require ISE AD join. Contingent on acquiring host/machine name from DHCP or DNS.
pxGrid	2	1	1	IoT Asset Attributes Custom Attributes	Requires external source that is pxGrid publisher. Custom attributes sent are determined by source, not ISE. Recommended for IoT devices and other sources of unique endpoint context.
Custom Attributes	2	1	1	Custom Attributes	Requires population of endpoint attributes via import, API, pxGrid, or manual entry. Useful for IoT or any endpoint where user-defined attributes can be extracted from external source and contribute to classification, trust, and state.

Enabling of probes is performed in ISE and coordinated with NAD configurations, and is described in detail in *ISE Profiling Design Guide* [9].

2.3.2 Profiling Endpoints Using Attributes Collected from Probes

Endpoint profiling can be classified into two primary approaches: dynamic and static profiling. Dynamic endpoint profiling relies on real-time analysis of device attributes and behaviors to assign profiles adaptively. In contrast, static endpoint profiling is manually configured by the administrator. If both methods are applied concurrently, static profiling overrides any dynamic profiling records. In this section we will be focusing mainly on dynamic endpoint profiling. [9]

Profiling of endpoint devices can begin after enough attribute data has been gathered by ISE probes. Profiling can be conducted using Cisco's prebuilt profiler policies or by defining customized rule sets.



Other Attributes	
161-udp	snmp-SNMPv1 server-public
ElapsedDays	140
EndPointPolicy	Unknown
EndPointProfilerServer	ise-psn-1.company.com
EndPointSource	SNMPQuery Probe
IdentityGroup	Unknown
InactiveDays	0
LastNmapScanTime	2018-Sep-04 10:42:26 UTC
MACAddress	00:09:E6:00:63:0F
MatchedPolicy	Unknown
NADAddress	172.16.1.67
NmapScanCount	3
NmapSubnetScanID	1536057587632
OUI	Cyber Switching Inc.
PolicyVersion	21
StaticAssignment	false
StaticGroupAssignment	false
Total Certainty Factor	0
ip	172.16.1.67
operating-system	Sensatronics EM1 environmental monitor (accuracy 96%)
operating-system-result	Sensatronics EM1 environmental monitor (accuracy 96%)
sysDescr	DUALCOM-S-16

Figure 10: Reviewing attributes from an unknown endpoint. [9]

A list of attributes is collected under each endpoint device's properties as seen in Figure 10. In the example, the author of Cisco ISE Profiling Design Guide has highlighted 161-udp, EndPointSource, OUI and sysDescr as valuable attributes that could be useful in matching the device to a profiler policy. [9]

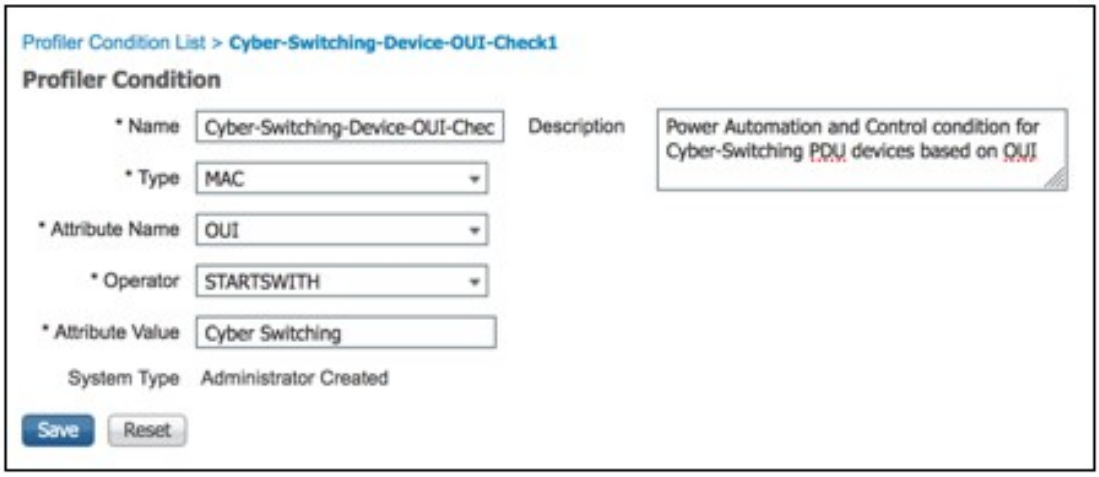


Figure 11: User-defined profiler condition. [9]

Inside of a profiler policy, a profiler condition checks a selected attribute using an operator whether that attribute matches a specified value or pattern. Table 5 lists every possible attribute that can be used in profiler conditions. In Figure 11 we can see an example of a custom profiler condition configured to evaluate if the OUI attribute string starts with ”Cyber Switching”.

Table 5: All available attributes for creating profiling conditions. [9]

Protocol	**Attributes**
DHCP	boot-file, client-fqdn, client-identifier, device-class, dhcp-class-identifier, dhcp-client-identifier, dhcp-message-type, dhcp-parameter-request-list, dhcp-requested-address, dhcp-user-class-id, dhcpv6-client-identifier, dhcpv6-ia-na, dhcpv6-ia-ta, dhcpv6-interface-id, dhcpv6-server-identifier, dhcpv6-user-class, dhcpv6-vendor-class, dhcpv6-vendor-opts, domain-name host-name, name-servers, pxe-client-arch, pxe-client-machine-id, pxe-client-network-id, server-identifier, vendor-class,
MAC	MACAddress, OUI

SNMP	cafSessionAuthorizedBy, cafSessionAuthUserName, cafSessionAuthVlan, cafSessionClientMacAddress, cafSessionDomain, cafSessionStatus, cLApIfMacAddress, cLApName, cLApNameServerAddress, cLApNameServerAddressType, cLApSshEnable, cLApSysMacAddress, cLApTelnetEnable, cLApTertiaryControllerAddress, cLApTertiaryControllerAddressType, cLApUpTime, cLApWipsEnable, cldcAssociationMode, cldcClientAccessVLAN, cldcClientIPAddress, cldcClientStatus, dot1xAuthAuthControlledPortControl, dot1xAuthAuthControlledPortStatus, dot1xAuthSessionUserName, hrDeviceDescr, hrDeviceStatus, ifDescr, ifIndex, ifOperStatus, port, portIfIndex, sysContact, sysDescr, sysLocation, sysName, sysObjectID, Vlan, VlanName, vlanPortVlan, vtpVlanIfIndex, vtpVlanName, vtpVlanState
IP	EndpointSource, FQDN, Host, ip, ipv6, mask, operating-system-result, PortalUser, User-Agent
RADIUS	Acct-Input-Gigawords, Acct-Output-Gigawords, Called-Station-ID, Calling-Station-ID, Chargeable-User-Identity, Connect-Info, Delegated-IPv6-Prefix, Delegated-IPv6-Prefix-Pool, Device-Type, DNS-Server-IPv6-Address, Egress-VLAN-Name, Egress-VLANID, Framed-Interface-Id, Framed-IP-Address, Framed-IP-Netmask, Framed-IPv6-Address, Framed-IPv6-Pool, Framed-IPv6-Prefix, Framed-IPv6-Route, Framed-Pool, Location, Location-Capable, Login-IPv6-Host, NAS-Filter-Rule, NAS-Identifier, NAS-IP-Address, NAS-IPv6-Address, NAS-Port, NAS-Port-Id, NAS-Port-Type, Route-IPv6-Information, Service-Type, Stateful-IPv6-Address-Pool, User-Name, VendorSpecific
NetFlow	agg_version, aggregation, CLASS_ID, count, DIRECTION, dOctets, dPkts, dst_as, DST_MAC, DST_MASK, DST_TOS, DST_VLAN, dstaddr, dstport, engine_id, engine_type, first, FIRST_SWITCHED, FLOW_SAMPLER_ID, flow_sequence, FLOWS, FragmentOffset, ICMP_TYPE, IN_BYTES, IN_PKTS, input, INPUT_SNMP, IP_PROTOCOL_VERSION, IPV4_DST_ADDR, IPV4_IDENT, PV4_NEXT_HOP, IPV4_SRC_ADDR, IPV6_DST_ADDR, IPV6_DST_MASK, IPV6_FLOW_LABEL, IPV6_SRC_ADDR, IPV6_SRC_MASK, L4_DST_PORT, L4_SRC_PORT, last, LAST_SWITCHED, MAX_PKT_LNGTH, MAX_TTL, MIN_PKT_LNGTH, MIN_TTL, nexthop, OUT_BYTES, OUT_PKTS, output, UTPUT_SNMP, prot, PROTOCOL, sampling_interval, source_id, src_as, SRC_MAC, SRC_MASK, SRC_TOS, SRC_VLAN, srcaddr, srcport, sys_uptime, tcp_flag, TCP_FLAGS, TOS, TOTAL_BYTES_EXP, TOTAL_FLOWS_EXP, TOTAL_PKTS_EXP, unix_nsecs, unix_secs, version

CDP	cdpCacheAddress, cdpCacheCapabilities, cdpCacheDeviceId, cdpCachePlatform, cdpCacheVersion
LLDP	lldpCacheCapabilities, lldpCapabilitiesMapSupported, lldpChassisId, lldpManAddress, lldpPortDescription, lldpPortId, lldpSystemCapabilitiesMapEnabled, lldpSystemDescription, lldpSystemName, lldpTimeToLive
NMAP	110-tcp, 123-udp, 135-tcp, 135-udp, 137-udp, 138-udp, 139-tcp, 139-udp, 143-tcp, 1434-udp, 161-udp, 162-udp, 1900-udp, 21-tcp, 22-tcp, 23-tcp, 25-tcp, 3306-tcp, 3389-tcp, 443-tcp, 445-tcp, 445-udp, 500-udp, 515-tcp, 520-udp, 53-tcp, 53-udp, 631-tcp, 631-udp, 67-udp, 68-udp, 80-tcp, 8080-tcp, 9100-tcp, operating-system, SMB.cpe, SMB.domain, SMB.fqdn, SMB.lanmanager, SMB.operating-system, SMB.server, SMB.workgroup
NMAP Extension	8081-tcp, <All Custom Ports>
Multimedia	H323DeviceName, H323DeviceVendor, H323DeviceVersion, mdns_VSM_class_identifier, mdns_VSM_srv_identifier, mdns_VSM_txt_identifier, sipDeviceName, sipDeviceVendor, sipDeviceVersion
ACIDEX	device-platform, device-platform-version, device-type
IOTAsset	assetDeviceType, assetHwRevision, assetId, assetIpAddress, assetMacAddress, assetName, assetProductId, assetProtocol, assetSerialNumber, assetSwRevision, assetVendor
ACTIVEDIRECTORY_PROBE	AD-Host-Exists, AD-Join-Point, AD-Operating-System, AD-OS-Version, AD-Service-Pack
CUSTOMATTRIBUTE	<All Custom Attributes>

Profiler conditions are used to collect Certainty Factor (CF), a cumulative integer value that is increased by matching conditions inside of a profiler policy. Each if-function in the profiler policy configuration is set to increase the CF by a defined value as seen in Figure 12. An endpoint is assigned the profile which receives the highest Total Certainty Factor (TCF) and meets the defined minimum CF. [9]

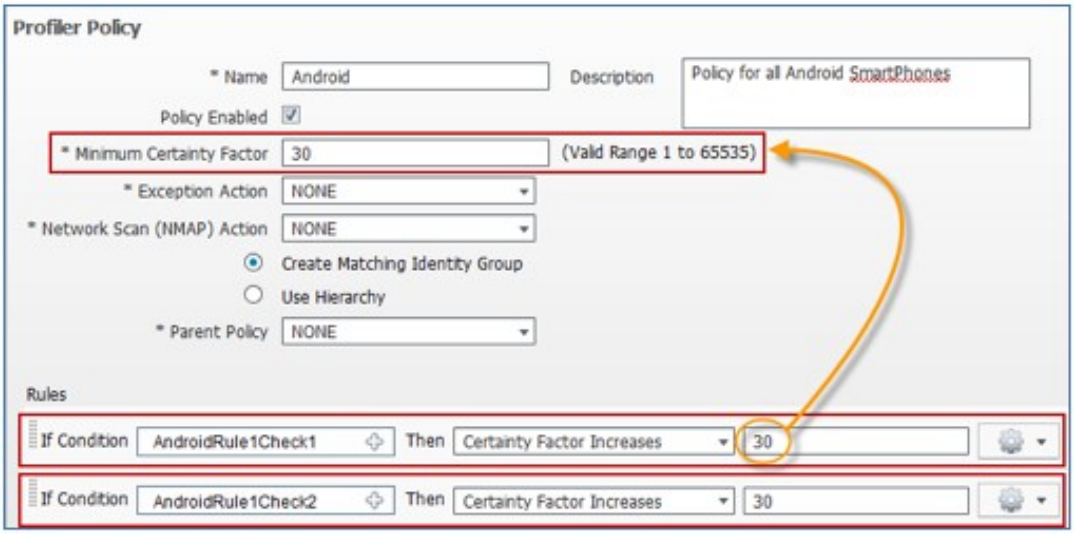


Figure 12: Profiler policy configuration window. [9]

As a general rule using predefined policies, Cisco recommends keeping CF values at their default settings, modifying them only when necessary to enforce a specific profile assignment. Cisco recommends creating a custom profiler policy based on the original, isntead of modifying the pre-existing policy, because automatic updates to the Cisco provided list of profiler policies might wipe out any changes made by the administrator. Best practice when creating custom profiler policies, is to assign similar CF weightings for the same types of attributes tested, maintaining consistent logic when calculating probabilities for a profile match. [9]

Hierarchical structuring of policies is used for systematization, and more importantly, inheritance purposes. For example in Figure 13, there is a parent policy named ”Apple-Device” based on an OUI condition, and then a child policy named ”Apple-iPhone” based on it’s own conditions narrowing down on details and ensuring more profiling accuracy. By using inheritance, redundant testing of same conditions for each device of similar type can be avoided.

Figure 13: Profiling policies in a hierarchical system. [9]

Logical Profiles allow administrators to group related profiles, for example ”iPhone” and ”Android Phone” under a unified group named ”Mobile Devices”, for more streamlined management. Configuration of a logical profile can be seen in Figure 14, where all power management and distribution devices are brought together under the same group.

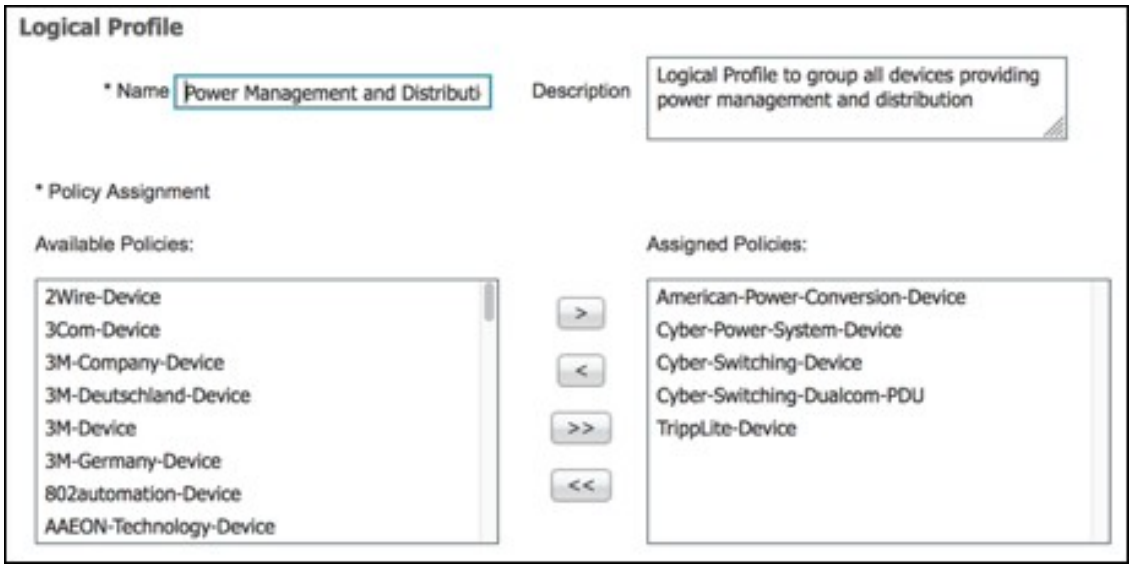


Figure 14: Logical Profile Configuration window. [9]

While conventional endpoint profiling methods rely on standardized attributes, IoT devices can sometimes lack these characteristics due to their proprietary architectures and limited protocol support. Consequently, accurate classification

may require the utilization of custom profiling attributes tailored to the unique operational signatures of such devices. [9]

2.3.3 Endpoint Custom Attributes

A custom attribute is a profiling parameter that falls outside the standard set of device-identifying characteristics, such as the OUI, sysDescr, or any other specified attribute listed in Table 5. Custom attributes are defined by administrators to capture unique identifiers or behaviors that are specific to certain classes of devices. Custom attributes are particularly relevant when profiling endpoints which lack conventional identifiers. A custom attribute might be derived from application-layer metadata, device's unique traffic patterns or configuration values. [9]

Practical applications demonstrate the necessity of custom attributes in environments where specialized devices are deployed. A custom attribute might, for example, be created to recognize a medical device that communicates exclusively through a proprietary protocol, or to identify a smart surveillance camera that periodically initiates DNS queries to a vendor-specific cloud service. In such cases, standard identifiers may be absent or insufficient, and custom attributes provide the distinguishing criteria required to correctly classify and manage the device. [9]

> Custom Attributes are a special use case. They are not a profiling result, but can be populated through ISE Profiling Services, such as the pxGrid probe. It is possible to use these Custom Attributes in the creation of Profiler Policy used to classify endpoints, but it is also possible to reference the Custom Attribute values directly in ISE policy rules. [9]

Custom attributes should be named carefully to prevent conflicts with existing attributes to ensure proper operation. Consistent naming conventions help ensure clarity, prevent conflicts, and make it easier to integrate and share these attributes across multiple systems. [9]

2.3.4 NetFlow

NetFlow is a Cisco IOS technology used to collect and monitor network traffic flows. These flows can be analyzed to detect abnormal communication patterns between devices. NetFlow also enhances endpoint device profiling when other sources of attribute data fail to identify a device. The fields monitored by NetFlow are listed in Table 6. NetFlow can be used as an ISE probe or paired with a NetFlow analyzer tool. As a probe NetFlow can be resource intensive if deployed on a large scale, hence targeted deployment is advised. [9][10]

Table 6: NetFlow IPv4 Supported Fields. [10]

Field	**Key or Nonkey Field**	**Definition**
IPv4 Protocol	Key	Value in the IPv4 protocol field.
IPv4 ToS	Key	Value in the type of service (ToS) field.
IPv4 Source Address	Key	IPv4 source address.
IPv4 Destination Address	Key	IPv4 destination address.
Transport Source-port	Key	Value of the transport layer source port field.
Transport Destination-port	Key	Value of the transport layer destination port field.
Interface Input	Key	Interface on which the traffic is received.
Counter Bytes	Nonkey	Number of bytes seen in the flow.
Counter Packets	Nonkey	Number of packets seen in the flow.

The attributes learned from the NetFlow probe are single valued and update the endpoint record with only the last values reported. Therefore, it is not currently possible to create a condition based on

> the detection of multiple values. For example, a Profiler Condition may be based on a specific TCP destination port or inclusion in a range of TCP ports, and it is possible to have a Profiling Policy based on a logical OR of such conditions, but it is not possible to have a Profiling Policy that matches multiple ports at the same time (a logical AND of protocols and ports). Furthermore, if the endpoint is profiled based on a matching protocol and port, and then different values are learned for the same endpoint, it is possible that the endpoint profile will continuously flap based on the last reported NetFlow export. This will result in potentially a huge volume of replication traffic to synchronize the latest profile and likely result in a repeated disruption in service for the endpoint. [9]

To avoid flapping the endpoint's profile, profiler Exception Actions can be used for a static profiler policy assignment. Exception Actions will be discussed in chapter 2.3.9. Once assigned a static profiler policy, the endpoint's profile will not change even if varying flow records are received from the NetFlow probe. [9]

NetFlow does not provide visibility inside the contents of network packets, as it is not a Deep Packet Inspection (DPI) method. However, Cisco does offer DPI solutions for endpoint device profiling and identification.

2.3.5 Deep Packet Inspection

Network-Based Application Recognition (NBAR) and Controller-Based Application Recognition (CBAR) are DPI solutions capable of inspecting network traffic up to OSI Layer 7 (L7). NBAR and CBAR can be used by Cisco's AI Endpoint Analytics. [11]

NBAR supports application recognition for up to 15,000 protocols. IoT, building automation and healthcare device protocols such as BACNET, MQTT, mDNS, DICOM and HL7. [11] NBAR supports provisioning of up to 450 interfaces on Cisco Catalyst 9000 devices. [12]

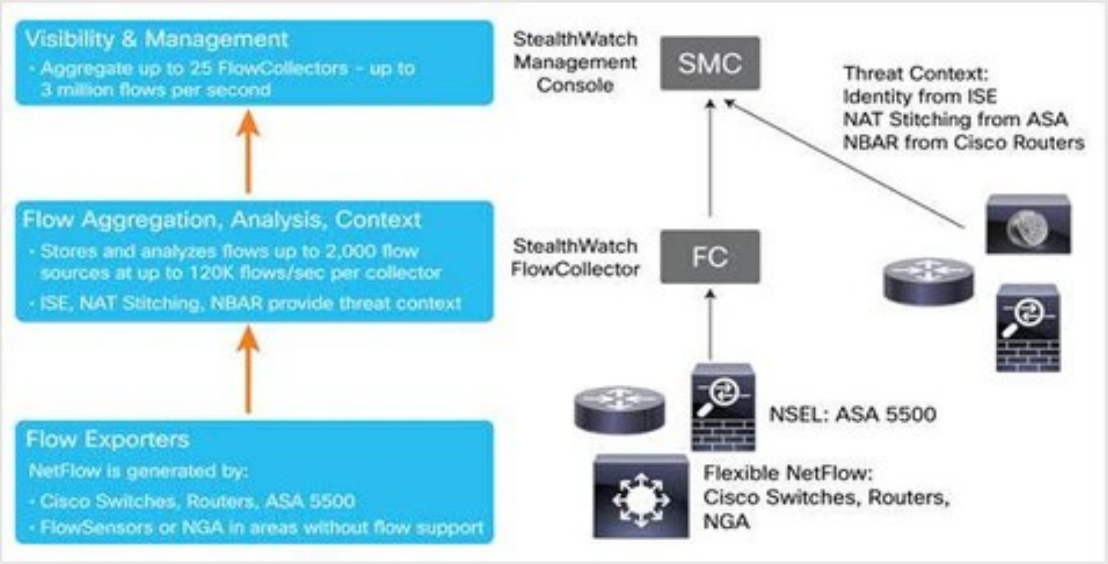


Figure 15: Cisco NBAR. [13]

NBAR signatures are local to the network device and need to be updated regularly by consuming NBAR Protocol Packs containing Protocol Description Language (PDL) files. CBAR aims to eliminate the need for this widespread management of signatures and solves the problem of classification of traffic that experiences assymetric routing. CBAR helps to keep the network up to date by identifying new applications as they continue to increase and allow updates to protocol packs. If Application Visibility is lost due to outdated protocol packs, applications may be incorrectly categorized. This will not only cause visibility holes within the network but also incorrect queuing or forwarding issues. CBAR solves that issue by allowing the push of updated protocol packs across the network. [14][15][16]

2.3.6 AI Endpoint Analytics

Cisco AI Endpoint Analytics is an endpoint visibility solution that helps to identify and to profile endpoints devices by enabling the assignment of Multi-Factor Classification (MFC) labels to endpoints (Figure 16), using the telemetry information received from various sources, including ISE probes and the NBAR/CBAR. [17]

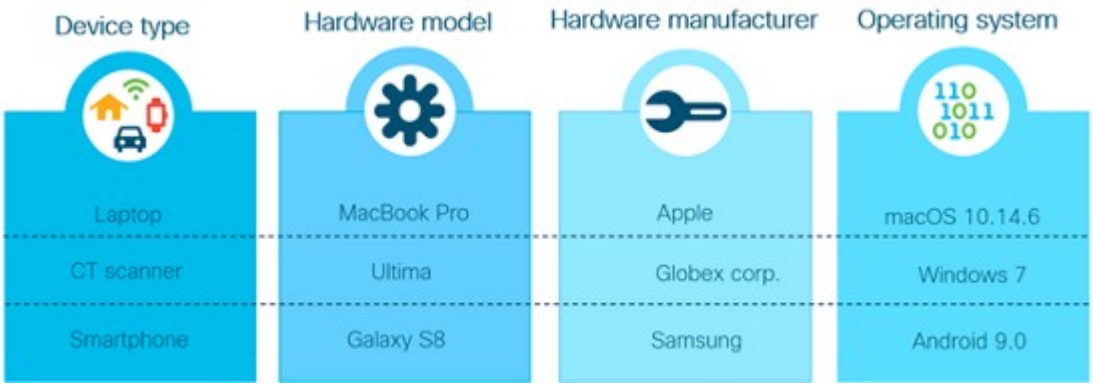


Figure 16: MFC labels assigned to endpoints. [11]

AI Endpoint Analytics sends the assigned MFC labels to ISE, where they can be used to create conditionals for custom endpoint profiler policies. The aggregation of data from different sources inside the network enables a more accurate and robust solution to profiling non-conventional endpoint devices. Cisco Platform Exchange Grid (PxGrid) enables the connection of Catalyst Center and ISE, allowing them to share contextual information in real time. [11]

2.3.7 Cisco Platform Exchange Grid

Cisco has developed a publisher/subscriber framework to help large and complex networks manage the growing variety of endpoint devices. The Platform Exchange Grid (pxGrid) provides a centralized solution for multivendor, cross-platform data exchange, enabling security, policy enforcement, and management control. [9][18]

In the ISE pxGrid node, pub/sub rights are managed by the pxGrid controller, which oversees published topics, authorizes subscribers, and acts as a ”switchboard operator” for establishing direct pub/sub communications [9]. Figure 17 illustrates the use of pxGrid. In this scenario, Cisco IND acts as the publisher, discovering information about IoT devices behind an industrial ethernet switch. The IND publishes the data that was collected to the pxGrid Subscriber through the pxGrid controller node. The subscriber is also capable of

performing bulk queries when required after an outage or on initial registration. [9]

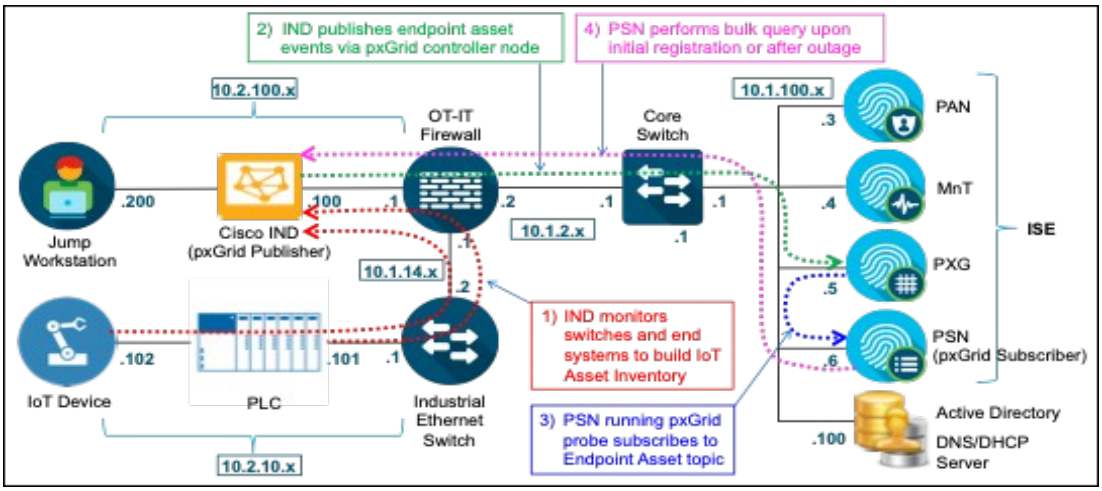


Figure 17: PxGrid probe in action. [9]

> PxGrid Topics refer to specific categories of information that can be shared between different Network Devices and Security Platforms through the pxGrid framework, allowing Contextual Data such as User Identity, Endpoint Details and Network Policies to be exchanged. [18]

2.3.8 Authorization of Profiled Endpoint Devices

After successful endpoint profiling, the final step is the evaluation of authorization policy. In Figure 18 we can see an example of an authorization policy configured. This policy dictates the level of network access granted to the connecting endpoint. The authorization determined by the endpoint's assigned profiler policy, its logical group membership or any other distinguishable property that can be used for identification.

We can see that Figure 18 example shows Identity Groups being used as conditions. Cisco does not recommend the use of Identity Groups when profiles and logical profiles are being used:

> The default setting in a new Profiler Policy is to create Matching Identity Groups. The general recommendation is to NOT create

new Endpoint Identity Groups based on Endpoint Profiles. Endpoints can be a member of only one Endpoint Identity Group and different ISE services (for example, device registration in Hotspot, Guest and BYOD flows) may overwrite this group object. Therefore, to avoid conflicts with other ISE services and potential explosion of new Identity Groups, the best practice is to match policy conditions on the Endpoint Profile or a Logical Profile when a decision needs to be made based on device profile. [9]

Figure 18: Authorization policy configuration window. [9]

Once the authorization policy has been evaluated, the corresponding access control decision of least privilege is enforced. This ensures that endpoints are granted access only to the network resources appropriate to their role and trust level. By dynamically applying segmentation and policy enforcement, the system not only minimizes the attack surface but also supports adaptive security allowing for real-time adjustments when endpoint attributes change. In practice, this approach embodies the principles of Zero-Trust, where no device is inherently trusted and access is continuously validated. [20]

2.3.9 Change of Authorizaiton

Because dynamic endpoint profiles are assigned based on active device attributes evaluated against profiler policy conditions, a device’s profile may change over time. When this occurs, the device’s permitted access must be re-

evaluated and updated to align with the applicable authorization policy. To enforce this adjustment, Cisco ISE initiates a Change of Authorization (CoA).

> CoA is a standards-based RADIUS feature (RFC 3576) that allows the RADIUS server (ISE) to initiate an unsolicited communication to the network access device (the RADIUS client) to update its access policy for an endpoint when certain state changes or policy changes occur. The update occurs without requiring that the endpoint initiate the reauthentication. [9]

Configuration of global profiler settings has an option to specify the CoA type (Figure 19), establishing the default behavior for how endpoints are reauthorized when their profiles change. Administrators can define whether profiling transitions trigger No CoA, Port Bounce, or Reauth at the global level. No CoA works well with visibility-only phase by preventing any reauthorization events. Port Bounce forces endpoints to go through a complete reauthorization cycle with IP refresh, making it the most disruptive but also the most reliable option for immediate policy enforcement. Reauth, on the other hand, is less disruptive and suited to scenarios where no VLAN or addressing changes are expected. On shared switch ports, ISE defaults to Reauth when a workstation connects through an IP phone to prevent service disruption. CoA type is also possible to configure on a per profile basis, taking precedence over any global settings. [9]

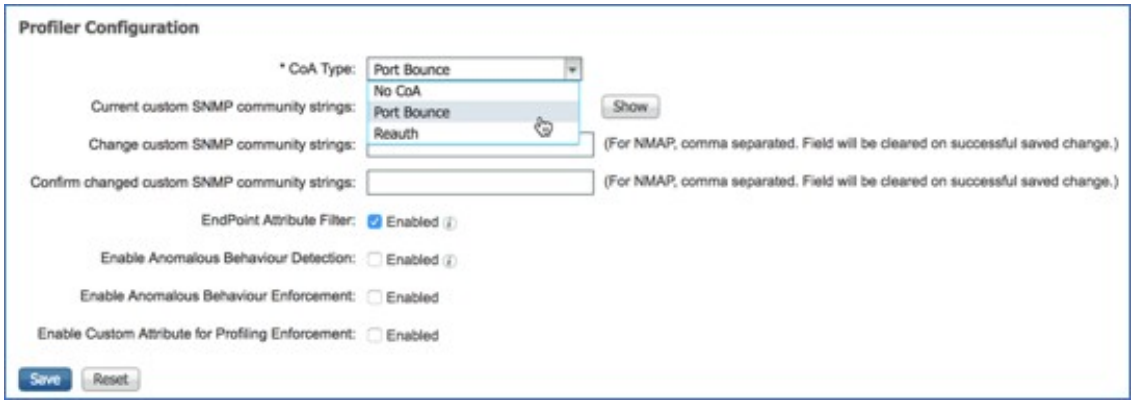


Figure 19: Global profiler configuration and CoA types. [9]

Exception Actions can be used to define what happens in the event of a profiling state change when certain conditions are met. ISE provides three built-in, predefined and non-configurable Exception Actions: AuthorizationChange (trigger CoA when a profile change yields a different authorization), EndpointDelete (trigger CoA when an endpoint is deleted or its profile becomes ”Unknown”) and FirstTimeProfile (trigger CoA when an endpoint’s profile changes from ”Unknown” to an actual profile). [9]

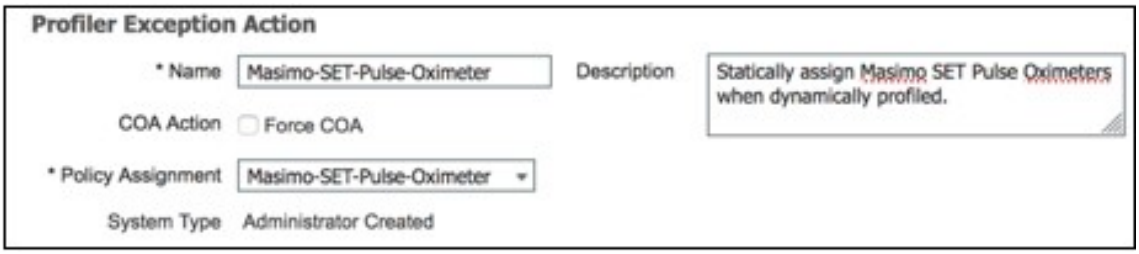


Figure 20: Custom Exception Action being configured with a policy assignment and no CoA. [9]

Besides triggering CoA, Exception Actions can also be used in a Profiler Policy to apply static policy assignments. Figure 20 shows the creation of a user-defined Exception Action that has CoA disabled and Policy Assignment enabled. Exception Action can be set to be triggered by fulfilling a condition in the profiler policy. [9]

> The use of Exception Actions can be a tool in cases where a static policy assignment needs to be made. Realize however, that once an endpoint is statically assigned to a profile, only an administrator can change that assignment. [9]

3 Profiling Strategy

Before any testing can be conducted, a profiling strategy must be carefully planned. It must be taken into consideration that all tests will be carried out in the target network's production environment, as no simulated test environment is available.

First the target network environment will be mapped. The relationships between network devices will be visualized as a network topology graph. All connected endpoints, together with their associated vendor names and VLAN IDs (VIDs), will be compiled into a summary table. VLAN assignments on the network will be graphed to help understand how the environment is segmented.

Once the mapping of the network environment is done and a profiling strategy has been established, the probes and other sources of attribute data needed for an accurate profiling of the target network environment's endpoint devices will be selected.

3.1 Mapping the Target Network Environment

Organization A has assigned me to a large office complex, providing a practical environment for implementing and testing endpoint profiling solutions. The environment hosts numerous workstation and non-workstation endpoint devices. The biggest challenge will be the successful profiling of miscellaneous IoT devices.

The dataset of all the endpoint devices connected to the target network environment was exported on 30.10.2025 as a CSV file from Cisco Prime Client and User section using a device name filter including items only from the target network environment to produce Table 7, Figure 22, and Figure 23. The total number of wired endpoint devices is 571. CDP neighbor data extracted from

Cisco Prime has been used to construct a network topology graph illustrated in Figure 21.

3.1.1 Network Topology

The mapping of the network environment will begin with drawing a network topology that includes all network switches and their connections seen in Figure 21. Figure 21 will be included as Appendix 1 to provide a clearer and more detailed viewing experience.

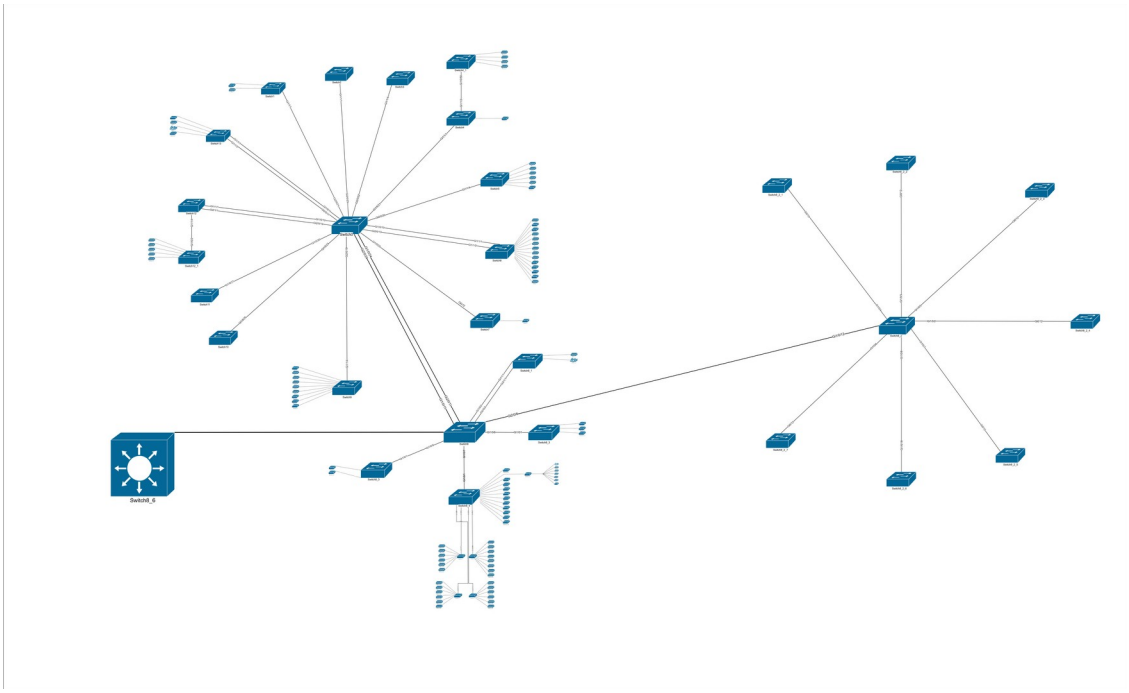


Figure 21: Target environment's network topology (Appendix 1). Device names have been redacted.

The environment is divided into two primary network segments. The first segment hosts mainly workstations, printers and other office devices. The second segment is isolated from the office network and dedicated to IoT and to building automation devices, which are largely managed by a single third party provider, Partner C.

3.1.2 Vendor Names

By examining the vendor names of connected endpoint devices, we can approximate all the different possible endpoint device types that would need to be profiled.

Table 7 lists each unique vendor name identified from the dataset, the number of occurrences indicating how many endpoints are associated with that vendor, the VLANs assigned between those endpoints, and the suspected endpoint types. The suspected endpoint types were inferred by analyzing information published by each vendor regarding their products and areas of specialization.

Table 7: Each unique vendor name paired with number of endpoints, VLAN IDs assigned, and suspected endpoint types. VLAN IDs have been redacted.

Vendor Name	**Number of endpoints**	**Associated VLAN IDs**	**Suspected Endpoint Type**
ASUSTek COMPUTER INC	1	3	Workstation
EliteGroup Computer	1	4	Workstation
Lantronix	1	6	IoT, Switching Device
LiteON	1	6	IoT
BIZLINK TECHNOLOGY	1	7	Interconnection, Adapter
HAKKO ELECTRONICS CO	1	7	IoT, HMI
LG Electronics	1	7	Smart TV
Pacom Systems Ltd.	1	7	IoT, Security
SHENZHEN JEHE TECHNO	1	7	IoT, Workstation
AirFiber Inc.	1	9	P2P Antenna
Armorlink Co .Ltd	1	9	IoT, Automation

Brother Industries	1	9	Printer
Elo touch solutions	1	9	POS System
Embedtronics Oy	1	9	IoT, Automation, Monitoring
ENGETRON-ENGENHARIA	1	9	UPS
Hewlett Packard	1	9	Workstation, Printer
HP Inc.	1	9	Workstation, Printer
Luxshare Precision I	1	9	Interconnection, Adapter
Toradex AG	1	9	IoT, Automation
Compulab Ltd.	1	10	IoT
Samsung Electronics	1	11	Smart TV
Atmel Corporation	1	12	IoT, HMI, Automation, Security
PEGATRON CORPORATION	1	12	IoT, Server, Storage
Super Micro Computer	1	12	Server, Storage, Switch
XIAMEN YEALINK NETWO	1	13	VOIP
AcSiP Technology Cor	1	15	Building Automation
DigiBoard	2	6, 9	Display Device
Micro-Star INTL CO.	3	4, 9	Workstation
Private	3	9	-
PRONET GMBH	3	12	Industrial Automation
AUDIO CODES LTD.	6	9, 14	VOIP
FIRST INTERNATIONAL	7	9	IoT, Automation, Building Automation
IEEE Registration Au	26	12	Partner C Device, Unknown

Raspberry Pi Foundat	26	12	Partner C Device, Unknown
Universal Global Sci	30	2, 6, 7, 8, 9	Access Point, IoT, Server, Workstation, Display, Medical, PoS
GOOD WAY IND. CO. L	34	2, 7, 9	Docking Station, Hub, Adapter, Interconnect, Video Capture, IoT, Monitoring
KONICA MINOLTA HOLDI	34	7, 9	Printer
Wistron InfoComm(Kun	35	5, 6, 7, 8, 9	Workstation, Docking Station, Server, Switching Device, VOIP, Display
LEXMARK INTERNATIONA	42	7, 9	Printer
Unknown	76	3, 5, 7, 8, 9	-
Cisco Systems Inc	83	1, 2, 7, 9, 14	Switching Device, Access Point, VOIP
LCFC(Hefei) Electron	135	6, 7, 8, 9	Workstation, Docking Station

Some vendor names clearly indicate the endpoint type they represent, while others are less straightforward. For example, vendors such as Konica Minolta and Lexmark can easily be associated with printers based on their names alone, although additional profiling attributes would be beneficial to confirm this assumption.

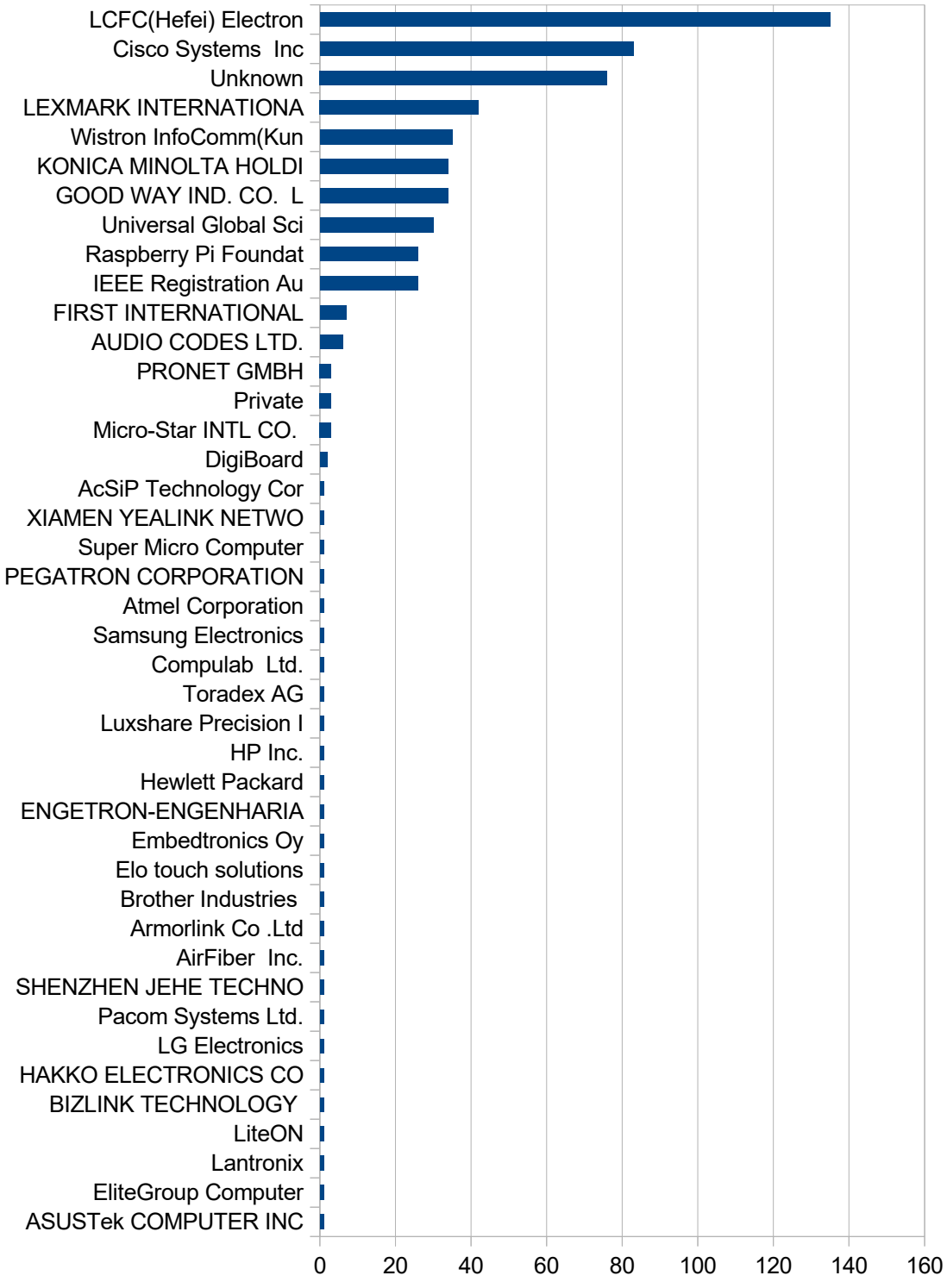


Figure 22: Distribution of device vendor occurrences presented in Table 7.

Raspberry Pi units and other unknown IoT devices can be noted. They are without documented ownership, assigned function or integration into

Organization A’s asset inventories. Insufficient visibility and accountability of these devices not only increase Organization A’s attack surface but also hinder effective migration to SDA, which relies on accurate device profiling and policy enforcement.

3.1.3 VLAN Distribution

Configured VLANs on the target network can be seen in Table 8. This data has been used in creating the Figure 23 pie chart. Adjustments have been made by grouping similar VLANs under the same name or giving them a more comprehensible description to make Figure 23 easier to read:

- ”wlanap1/2” (1, 2) have been combined to represent the ”Access Points” VLAN.
- ”customers1/2” (3, 4) and ”customers_wlan” (5) have been combined to represent the ”Guest” VLAN.
- ”workstations1/2/3/4/5” (6, 7, 8, 9, 10) have been combined to represent the ”Workstations” VLAN.
- ”voip1/2” (13, 14) have been combined to represent the ”VOIP” VLAN.
- ”partner” (12) has been renamed as ”External Network (Partner C)”.
- ”iot” (15) has been renamed as ”IoT”.
- ”smartTV” (11) has been renamed as ”Smart TV”.

Table 8: Target Network Environment's VLANs. VLAN descriptions and VLAN IDs have been redacted.

VLAN description	**Number of Endpoints**	**VLAN ID**
wlanap1	44	1
wlanap2	31	2
customers	3	3
customers2	3	4
customers_wlan	3	5
workstations1	15	6

workstations2	194	7
workstations3	30	8
workstations4	179	9
workstations5	1	10
smartTV	1	11
partner	58	12
voip1	1	13
voip2	7	14
iot	1	15

By examining the network's distribution of VLANs in Figure 23, we can observe that most endpoint devices are assigned to the "Workstations" VLAN group, which should be used for organizational workstations, docking stations, printers, and IP phones. Table 7 shows us that there could also be IoT and other non-conventional endpoint devices assigned to this VLAN group.

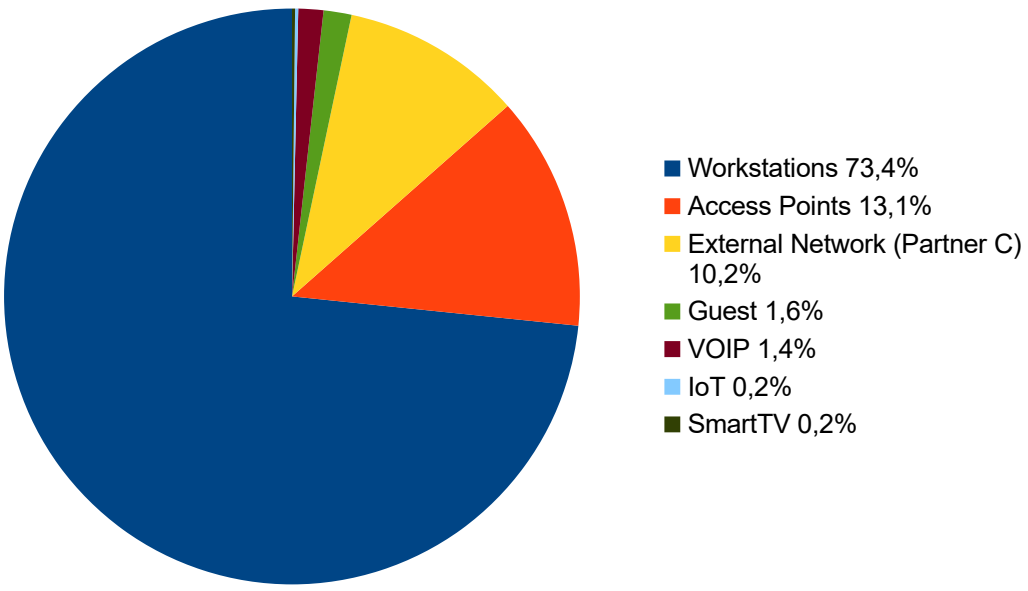


Figure 23: Distibution of VLANs on the target network environment.

The second largest VLAN group, “Access Points”, is designated for wireless access points used by social and healthcare services. The majority of connected devices within this group are manufactured by Cisco Systems Inc. (Table 7), reflecting the organization’s primary vendor for wireless infrastructure. However, a small number of devices are registered under two other vendor names, suggesting the possible presence of non-access point devices or equipment from alternative manufacturers.

The third most utilized VLAN, “External Network (Partner C)” is allocated to Partner C’s devices. As shown in Table 7, most devices within the VLAN are associated with IoT, automation, security, or server-related functions. Identifying and profiling these non-conventional endpoint devices within this VLAN is expected to present the greatest challenge due to their diversity and specialized roles.

3.2 Probe Selection

A critical requirement for the initial endpoint profiling stage is the non-disruptive collection of profiling attributes. Probes deployed must avoid causing significant network overhead and ensure that performance and service availability remain uncompromised throughout the discovery process. Device Sensor probing has been selected as the initial method for collecting profiling attributes, providing enough attribute data for basic endpoint devices and a foundation for more specialized probing techniques tailored to specific hard-to-profile targets. Probing using device sensor should be done in Monitor Mode with CoA disabled for least disruption to user experience caused by authenticaiton errors and profile changes.

A logical starting point for the profiling process is the profiling of endpoints that are relatively easy to identify, such as organizational workstations, printers and ip phones. After successfully profiling the more conventional endpoint types, the focus should shift to IoT and other non-conventional endpoint devices. Collecting the necessary profiling attributes for these devices is expected to

require greater effort and incur higher performance overhead to achieve accurate results.

Conducting tests in carefully controlled, small-scale phases with a limited number of endpoints and endpoint types enables systematic evaluation of probe efficiency, profiling accuracy, and the resulting impact on network performance and availability prior to large-scale deployment.This approach reduces the risk of inadvertently disrupting large groups of users. In line with risk management principles, it is preferable that any misclassifications or access-control errors affect a small subset of non-critical devices rather than critical systems, where operational continuity is paramount.

The probes currently enabled in ISE include RADIUS, SNMP Query, DHCP IP Helper, and the Active Directory probe. Additionally, NMAP is utilized during initial scans to provide basic visibility over all connected endpoint devices.

3.2.1 Device Sensor Probe for General Profiling

Two Catalyst switches are being considered for the initial test run. Configurations to ISE will be ordered from Partner B and the NADs will be configured by the Organization A supervisor. Device sensor probing will be tested in Monitor Mode.

Table 9: Device Sensor - Catalyst Supported Platforms. [19]

Platform	**License Level**	**Miniumum Version for Device Sensor (CDP/LLDP)**	**DHCP Glean**	**HTTP Glean**
3850	Lan Base	3.6.0E / 16.3.X	Wired & Wireless	wireless with Cisco AP From 3.6.1; no filter on switch
3850	IP Base	3.6.0E / 16.3.X	Wired & Wireless	wireless with Cisco AP From 3.6.1; no filter

				on switch
3850	IP Services	3.6.0E / 16.3.X	Wired & Wireless	wireless with Cisco AP From 3.6.1; no filter on switch
3650	Lan Base	3.6.0E / 16.3.X	Wired & Wireless	wireless with Cisco AP From 3.6.1; no filter on switch
3650	IP Base	3.6.0E / 16.3.X	Wired & Wireless	wireless with Cisco AP From 3.6.1; no filter on switch
3650	IP Services	3.6.0E / 16.3.X	Wired & Wireless	wireless with Cisco AP From 3.6.1; no filter on switch
3750X/ E/ V2 3560 X /CX /V2 2960 X/XR	Lan Base	15.0(1)SE	Wired	Not Supported
	IP Base	15.0(1)SE	Wired	Not Supported
	IP Services	15.0(1)SE	Wired	Not Supported
4500-E Sup 7E/7LE/ 8E	Lan Base	3.6.0E	Wired	Not Supported
	IP Base	3.6.0E	Wired	Not Supported
	Enterprise Services	3.6.0E	Wired	Not Supported
4500 Sup 6/6LE 4900 M/EF/E 4948 E/EF	Lan Base	15.1(1)SG	Not Supported	Not Supported
	IP Base	15.1(1)SG	Not Supported	Not Supported
	Enterprise Services	15.1(1)SG	Not Supported	Not Supported
Catalyst 6K Sup2T/6T * Only on IA Ports	IP Base	15.2(1)SY	Supported	Not Supported
	IP Services	15.2(1)SY	Supported	Not Supported
	Advanced IP Services	15.2(1)SY	Supported	Not Supported
	Enterprise Advanced Services	15.2(1)SY	Supported	Not Supported
Catalyst	Network	16.5.1a	Supported	Wireless with Cisco

9000 Family	Essentials			AP From 16.5.1a; no filter on switch
Catalyst 9000 Family	Network Advantage	16.5.1a	Wired & Wireless	Wireless with Cisco AP From 16.5.1a; no filter on switch
Catalyst vWLC, 8500, 8510 , 7500, 5508, WiSM2 WLC	?	AireOS 7.3.101.0	Yes	Yes
Catalyst 2504 WLC	?	AireOS 7.4.100.0	Yes	Yes
Catalyst 3504 WLC	?	AireOS 8.5.103.0	Yes	Yes
Catalyst 9800 WLC	?	IOS-XE 17.2.1	Yes	Yes

Organization A's Catalyst switches are running the appropriate software version supporting CDP, LLDP, and DHCP options for device sensor probing. This has been confirmed using Table 9 provided by Cisco. DHCP option support is enabled through the DHCP Snooping feature on the switch. The detailed configuration procedure is described in the *Cisco ISE Profiling Design Guide* [9].

3.2.2 AI Endpoint Analytics for IoT Device Profiling

AI Endpoint Analytics has been proposed as the solution for profiling IoT and other non-conventional endpoint devices. To enable its implementation, Partner B must prepare and configure Cisco ISE accordingly. Additionally, the Organization A supervisor would be responsible for ensuring that all necessary configurations supporting ISE node and Catalyst Center node connectivity are properly applied to facilitate AI Endpoint Analytics integration.

The goal of this test would be to get as accurate MFC labels as possible. The initial AI Endpoint Analytics results should be verified on-site by inspecting the physical devices for markings or other indicators of their function to confirm the accuracy of the generated labels. If on-site observations do not yield sufficient information, verification can be attempted through Partner C contacts. If the findings confirm the AI Endpoint Analytics profiling suggestions, future results can be regarded as more reliable.

While this method of collecting profiling attributes is probably the most effective, it may also pose the greatest challenges in terms of implementation.

3.2.3 NetFlow Probe for IoT Device Profiling Support

NetFlow has been selected to support IoT device profiling. Initial testing should focus on individual endpoints to minimize potential performance overhead from collecting and exporting NetFlow data.

Partner B will be preparing ISE for receiving NetFlow data from the NAD that will be configured by the Organization A supervisor. The NAD selected is a switch connecting Partner C’s devices to the network.

The goal is to test if NetFlow probing can advance the identification and profiling of IoT and non-conventional endpoint devices by analyzing protocols and ports used between these unknown devices. The responder and initiator IP addresses will be useful in mapping the relationships between these devices.

4 Results

4.1 Profiling Tests

It can generally be observed that each endpoint device profiled in these tests receives attributes from at least the NMAP and RADIUS probes, while some devices also obtain additional attributes from the AD and DHCP probes.

4.1.1 Test 1 – Device Sensor (2.9.2025)

The initial test run was conducted on the two Catalyst switches. Most endpoints connected to these switches received a profile, but some issues were encountered.

It was observed that the device-sensor cache on the switches was not recording the expected data, indicating a possible issue in the device-sensor configuration. A review of the running configuration revealed that DHCP snooping and LLDP were not enabled, both of which are required for the full functionality of the device-sensor probe. Additionally, one of the switches had TLV filter lists configured for CDP, LLDP, and DHCP attributes, while the other did not.

Despite the shortcomings in the device-sensor configuration, successful profiling of Windows 10 and 11 workstations can still be observed. It appears that attribute data received from the DHCP IP helper probe completes the profiling process for these workstations. The specific DHCP attribute involved is option 60 (Vendor Class Identifier). The profiling sequence for a Windows 10 or 11 workstation is as follows: Workstation → Microsoft Workstation → Windows 10/11 Workstation.

In cases where DHCP attributes are unavailable, the Username attribute obtained from RADIUS or AD probe could be used for profiling. Each workstation in Organization A is assigned a username that can serve as an

identifiable attribute for recognizing the endpoint as a workstation. All organizational workstations run the Windows operating system, but the AD username alone cannot distinguish between Windows 10 and Windows 11. The operating system version could provide valuable information for both monitoring and network segmentation.

An interesting case occurred with an HP device that had an AD workstation username but was placed in the logical profile group for printers. Normally, such classification would not raise concerns, except for the fact that the endpoint had an AD username associated with an organizational workstation. Upon investigating why ISE labeled this endpoint as a printer, it was discovered that the profiler policy of HP device relied on the device's OUI, which referenced Hewlett-Packard. Based on this match, a logical profiler policy automatically assigned the device to the printer group. This misclassification occurred due to insufficient attribute data. If ISE had received DHCP attributes collected by the device sensor, the issue could probably have been avoided.

It is suspected that the device-sensor probe is not functioning as intended and that the switch configuration may contain errors. The attribute data indicates that the last probe to report to ISE was the RADIUS probe, but it does not specify whether Device Sensor was functioning correctly. ISE is receiving data from the RADIUS probe, which suggests that the underlying framework supporting the device sensor is operating correctly. Both the RADIUS probe and the device sensor probe functionality rely on RADIUS accounting packets transmitted from the switch to the ISE PSN.

4.1.2 Test 2 – Device Sensor (11.9.2025)

Two Catalyst switches (one a one-stack switch and the other a two-stack switch) were introduced in the second test. These switches have a little more endpoint devices and variety among them. This test has the same goals and methods as the first test: to collect endpoint attribute data using device-sensor probe and see how much can be done with the attribute data collected.

We start our observations from the switches' device-sensor caches. Switch1 has one record in the cache that is another switch connected to its Gi1/1/1 port. The data that has been gathered is CDP. Switch2 has more records inside its device-sensor cache, but at further inspection, they appear to be access points. Therefore it can be observed that neither switch is recording any non-switch devices in their device-sensor cache.

Inspecting the running configurations of these switches, we can observe that LLDP and DHCP-Snooping are disabled on Switch1. On Switch2 LLDP is enabled but DHCP-Snooping is disabled.

The profiling results in this test have nine new notable categories of entries:

1. Nortel Device
   Most likely an IP phone, that has a workstation connected to it. The existence of a workstation has been deducted from the fact that the endpoint has an AD Username of a registered organizational workstation. To confirm this hypothesis, we would need to configure a Multi-VLAN Access Port (access/voice) to extract attribute data from both the IP-Phone and the Workstation. [20]

   Inspecting the running configuration of a switch hosting an example Nortel Device, we can see that the port of Nortel Device is configured with only an access VLAN, with no voice VLAN specified. Fixing this could potentially improve profiling accuracy.

   To more accurately profile the IP phone that received the Nortel Device profile and specify its model number, we need to obtain a minimum CF value of 20 by either matching the LLDP:lldpCacheCapabilities (20 CF) or DHCP:dhcp-class-identifier (20 CF) attributes. This again highlights the need to get the device sensor functioning properly in order to advance our profiling goals.

2. Lexmark Device
   The MFC profiler result for these devices is a printer. The profiler policy has gotten stuck at the Lexmark Device profile although it is quite clear that this is in fact a printer.

   For a successful profiling of Lexmark Printer we would need a minimum CF of 20 by either matching the SNMP:hrDeviceDescr (20 CF) or DHCP:dhcp-class-identifier (20 CF) attributes.

   If DHCP or SNMP attributes cannot be collected, an alternative solution would be to use the MFC profiler result as a custom attribute condition within a customized Lexmark Printer profiler policy.

3. Audio Codes Device
   This is most likely an admin created ”Audio Codes IP phone” profile, that checks the endpoint attribute DHCP:dhcp-class-identifier.

   The default ”Audio-Code-IPPhone” profile checks for LLDP:lldpSystemDescription and LLDP:lldpSystemName endpoint attributes each rewarding 20 CF towards the minimum CF of 20. For a more accurate device model defining profile DHCP:dhcp-class-identifier would have to be checked.

4. Axis-Device
   According to the manufacturer, Axis Communications, this is most likely a security camera. MFC Endpoint Type has been given the value ”IoT Device”.

   A profile ”Axis-Network-Camera” checks for the endpoint attributes of DHCP:dhcpv6-vendor-class and DHCP:dhcp-class-identifier, each rewarding 20 CF towards the minimum CF of 20.

For a more accurate profile involving the camera’s model, endpoint attribute of LLDP:lldpSystemDescription is required.

5. Konica-Minolta-Printer
   Profiles as it should. Gets the endpoint attribute of SNMP:sysDescr.

6. Cisco-Device
   Most likely an Access Point. CDP attributes from the device-sensor probe would be helpful here for a more defined profile.

7. Unknown (AD-Username)
   Appears to be an organizational workstation based on OUI and the AD-Username. For a more defined profile, DHCP endpoint attributes are needed. Alternatively, we could create a custom profiler policy that checks for the AD-Username.

8. Unknown (blank)
   The device’s OUI is ”Pacom Systems Lts”, which is a security solutions company. Data from a DPI process would benefit the profiling of this endpoint device.

9. Asus-Device
   Most likely an organizational workstation. MFC Endpoint Type has been given the value of ”Workstation”. More defined profiling requires the DHCP endpoint attributes or alternatively the use of AD-Username in a custom profiler policy.

The profiles listed above represent new additions to the collection of identified profiles, extending the results of Test #1 by introducing unique endpoint device profiles.

4.1.3 Test 3 – Device Sensor (22.9.2025)

Four new two-stack Catalyst switches were introduced for testing. This test is a continuation to the first two device-sensor tests conducted. All four switch stacks collect CDP, LLDP and DHCP data inside the device-sensor cache. The problem with connectivity between device-sensor cache and ISE still remains.

A new profile has been introduced named ”LG-Device”. MFC Endpoint Type has been given the value of ”Mobile Device”. The endpoint has been authenticated through WiredMAB. The odds of this device being a mobile device, like a smartphone, are pretty slim. The inaccurate profiling appears to be a result of insufficient endpoint attribute data. Finding out the device’s operating system through DHCP attributes would be beneficial for more accurate profiling. If the device doesn’t use DHCP, the SNMP or HTTP attributes could be of benefit instead. The identity of the LG-Device was later confirmed by an Organization A expert to be a smart TV.

After this test, it was again observed that the endpoint attributes lacking from the device-sensor probe are indeed seriously inhibiting the accumulation of accurate profiling results. Before further deployment, the device-sensor probe needs to be correctly configured.

The switches that are configured correctly for device sensor are gathering CDP, LLDP and DHCP attributes to the device-sensor cache, but the cached data is not received by ISE. This issue requires further investigation.

4.1.4 Test 4 – NetFlow (15.10.2025)

A NetFlow configuration was administered on a Catalyst switch with the goal of exploring the NetFlow probe for the means of supporting the profiling of unknown IoT devices.

A flow monitor and exporter were set up to send ISE the NetFlow data collected from one specific endpoint device connected to the switch. The monitor was set up with a 60-second active, and a 15-second inactive timeout setting. The data in NetFlow monitor cache is sent to the collector every 60 seconds if the flow is active and every 15 seconds if it is inactive. When the cache is sent to the collector, it is wiped from the monitor's cache.

In the monitor cache we can find an entry with an initiator being our endpoint tested. Analyzing the cache, we find out that the initiator from the local switch is connecting to an external switch in the Organization A's IoT network, operated by Partner C, four hops away. Responder's port is TCP 8888, which is not an official port assigned by the Internet Assigned Numbers Authority (IANA). The endpoint device could be communicating with a server. The initiator has the OUI of Partner C.

The objective of this test was to collect NetFlow data and send it to ISE to help with profiling IoT endpoint devices. The collection and exporting of NetFlow data appears to be working as intended. However, the endpoint monitored is not visible to ISE, which means that we can't view or use the data received by ISE towards the profiling of the endpoint. This problem might be caused by a missing MAC-to-IP binding that would link the NetFlow data received by ISE to the endpoint device which it belongs to. This would be solved by initiating a NMAP, RADIUS, DHCP or SNMP scan by ISE. [9]

After resolving the MAC-to-IP binding issue, the endpoint was successfully discovered by ISE. However, the endpoint's attribute list shows no records of NetFlow data. This indicates a deeper issue with the NetFlow-to-ISE connectivity. Further investigation is recommended by checking the PSN at the CLI level to verify whether any packets are being received from the NetFlow-enabled switch.

There could be a link between NetFlow data not being directed to the endpoint's attribute list and device sensor data not being received.

5 Conclusions

The RADIUS probe appears to be functioning correctly, and the probed endpoint devices are receiving attribute data as expected. The issue is that device-sensor attributes are not being recorded for the same endpoint devices that successfully receive the normal RADIUS probe data. Both the RADIUS probe and the device sensor use RADIUS accounting packets to transmit data to ISE. The absence of Device Sensor attribute data (DHCP, LLDP, CDP) inhibits the proper profiling of devices associated mainly with the “Workstations” VLAN, such as workstations, ip-phones, docking stations and printers. Missing device-sensor data also limits the effectiveness of AI Endpoint Analytics for future profiling of IoT and non-conventional endpoint devices by reducing the amount of information available for analysis.

Attributes from the DHCP, LLDP and SNMP protocols would have benefitted the accurate profiling of almost every endpoint device included in the device-sensor tests conducted. For example, the dhcp-class-identifier attribute extracted manually from a device-sensor-configured NAD was found extremely helpful in identifying an endpoint device with the vendor name of ”Audio Codes Device” as a AudioCodes MediaPack MP-112-FXS Analog VoIP Gateway.

Some unused but potentially valuable attribute data is associated with unprofiled devices that are probably workstations. The RADIUS/AD probe provides the “User-Name” attribute, which identifies each device’s registered AD username. In Organization A’s AD database, only workstation accounts are assigned names that begin with specific character sequences that follow Organization A’s naming standard. This information could be utilized in creating custom profiling policies for more accurate identifying of organizational workstations.

The primary focus when moving forward should be on solving the device-sensor functionality. While RADIUS accounting packets are being transmitted correctly,

as discussed earlier, they currently do not include device-sensor data or the data is being lost in transmission. The device-sensor data is stored in the NAD's device-sensor cache, where it can be viewed through the CLI and even used for manual device identification. The key challenge, therefore, is determining how to transmit and pair the cached device-sensor information with each corresponding endpoint device registered to ISE.

Once the issue with device sensor is resolved, attention can be directed toward further testing of the NetFlow probe and the initial testing of AI Endpoint Analytics. The concrete implementation of these methods across the entire target network environment should only be recommended after all profiling tests have been proven successful.

References

1 Nieminen, Mikko. 2023. Building Scalable LAN Solution. Master's Thesis Metropolia University of Applied Sciences. Theseus Database.

2 Wefeng, Xia. Yonggang, Wen. Chuan, Heng Foh. Dusit, Niyato. Haiyong, Xie. 2015. A Survey on Software-Defined Networking. Institute of Electrical and Electronics Engineers. IEEE Xplore.

3 Open Networking Foundation. 2016. SDN Architecture Issue 1.1 ONF TR-521. Open Networking Foundation.

4 Cisco. 2025. Cisco Software-Defined Access Solution Design Guide. cisco.com.

5 Utriainen, Joonas. 2019. Software-Defined Access. Bachelor's Thesis. XAMK. Theseus Database.

6 Network Lessons. Cisco Locator ID Separation Protocol (LISP). networklessons.com

7 Cisco. 2023. Cisco Catalyst Center At-a-Glance. cisco.com

8 Cisco. 2025. Cisco ISE Licensing Guide. cisco.com

9 Hype, Craig. 2025. ISE Profiling Design Guide. cisco.com

10 Cisco. 2021. Overview of Netflow. cisco.com

11 Cisco. 2021. Cisco AI Endpoint Analytics Deployment guide. cisco.com

12 Cisco. 2025 Cisco Catalyst Center User Guide, Release 2.3.7.x, Chapter: Provision Services. cisco.com

13 The Network DNA. 2020. All about Cisco NBAR. thenetworkdna.com

14 Cisco. 2011. NBAR Protocol Pack. cisco.com

15 Baldwin, Keith. 2024. Cisco Catalyst Center Template Labs. cisco.com

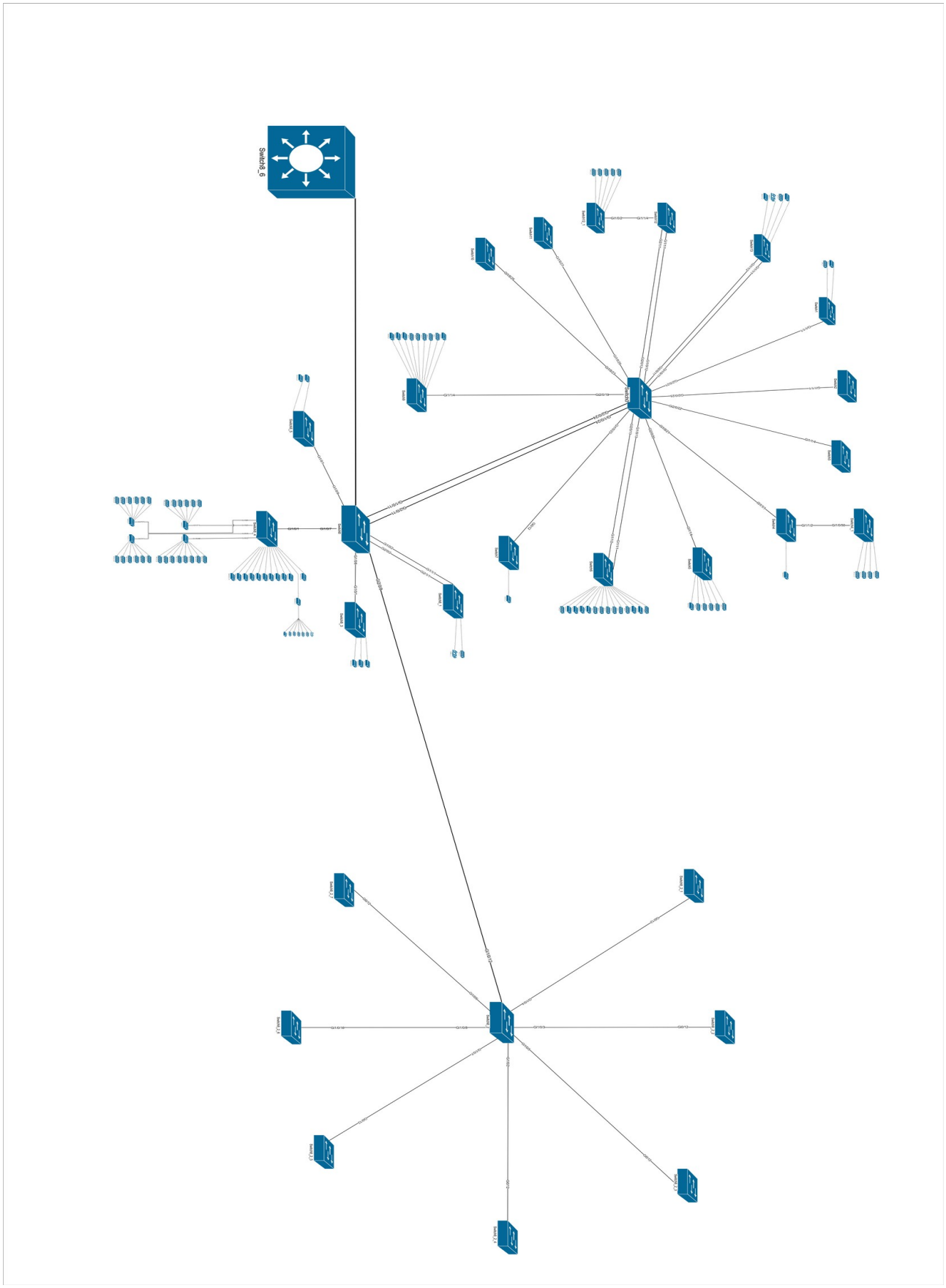
16 Chilcote, Preston. 2025. NBAR and CBAR. blogs.cisco.com

17 Cisco. 2025. Cisco Catalyst Center User Guide, Release 3.1.x, Chapter: Cisco AI Endpoint Analytics. cisco.com

18 Morais, Marcelo. 2025. ISE, What we need to know about pxGrid. cisco.com

19 Thomas. 2020. Device Sensor, Catalyst Supported Platforms. community.cisco.com

20 Network Lessons. Cisco Switch Port Configuration for IP Phone. networklessons.com


Switch8_9