

Chapter 25: Secure Network Access Control

Chapter 26: Network Device Access Control and Infrastructure Security

Instructor Materials

CCNP Enterprise: Core Networking



Network Security Design for Threat Defense

- Endpoints are extremely vulnerable to security threats, and they can become infected. A solid network security design protects the endpoints from these types of security threats and enforces endpoint network access.
- This chapter describes the components of network security design for a campus environment that are used to protect, detect, and remediate security threats and attacks.

Cisco SAFE

To address the evolving cybersecurity threats, Cisco created Cisco SAFE, a security architectural framework that helps design secure solutions for the following places in the network (PINs):

- Branch
- Campus
- Data Center
- Edge
- Cloud
- Wide Area Network (WAN)

Cisco SAFE focuses on the integration of security services within each of the PINs. For information on the underlying networking design and infrastructure see the Cisco Validated Design (CVD) guides, which provide detailed networking design and implementation guidance. CVDs can be found at www.cisco.com/go/cvd.

Network Security Design for Threat Defense

Cisco SAFE Domains

Cisco SAFE also defines secure domains, which are operational areas used to protect the different PINs. The following security concepts are used to evaluate each PIN:

- Management
- Security Intelligence
- Compliance
- Segmentation
- Threat Defense
- Secure Services

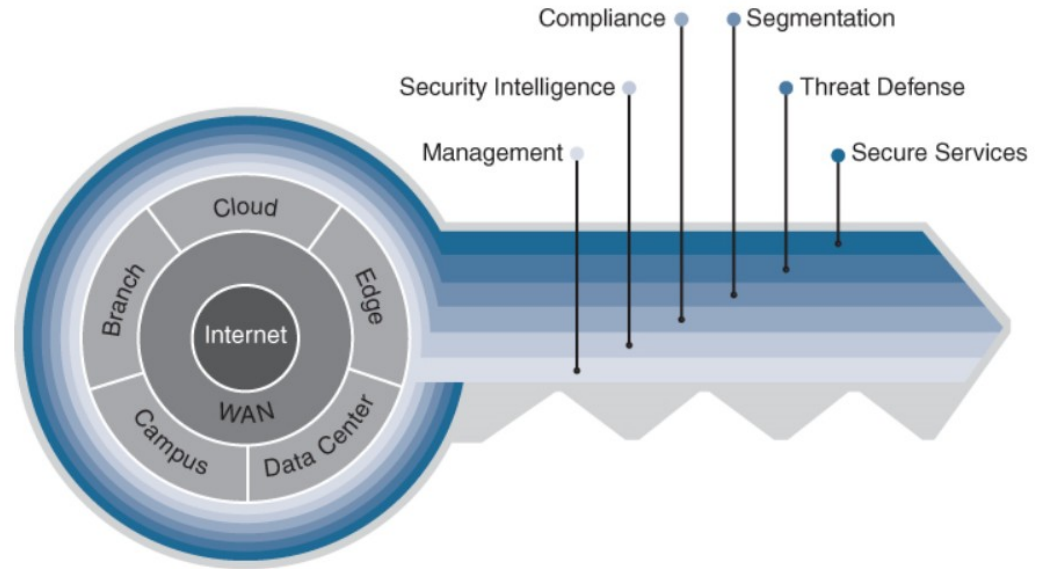


Figure 25-1 *The Key to Cisco SAFE*

Cisco SAFE Implementation

Implementing the Cisco SAFE framework in an organization provides advanced threat defense protection that spans the full attack continuum before, during, and after an attack for all the PINs:

- Before
- During
- After

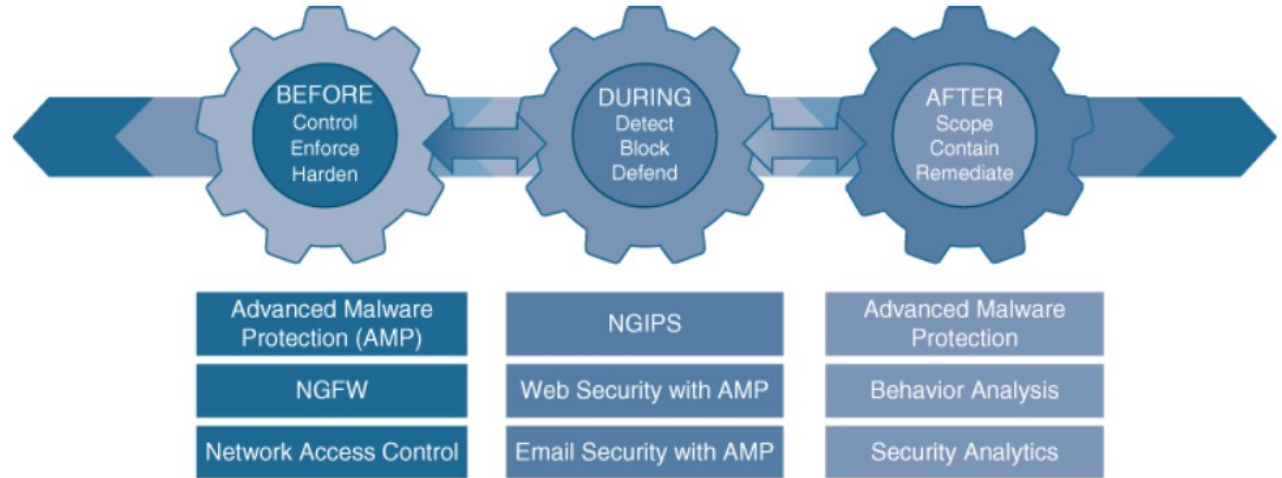


Figure 25-2 *Cisco Products and Solutions Across the Attack Continuum*

Next-Generation Endpoint Security

- To be able to detect the rapidly evolving threats, organizations should design their networks using a security framework such as that provided by Cisco SAFE.
- The following sections describe the most critical components needed to implement the Cisco SAFE framework for a campus environment (or *PIN*, in Cisco SAFE terminology).

Cisco Talos

Talos is the Cisco threat intelligence organization, an elite team of security experts who are supported by sophisticated security systems to create threat intelligence that detects, analyzes, and protects against both known and emerging threats for Cisco products.

Cisco Talos was created from the combination of three security research teams:

- IronPort Security Applications (SecApps)
- The Sourcefire Vulnerability Research Team (VRT)
- The Cisco Threat Research, Analysis, and Communications (TRAC) team

Talos receives valuable intelligence that no other cybersecurity research team can match through the following intelligence feeds:

- Advanced Microsoft and industry disclosures
The Advanced Malware Protection (AMP) community
- ClamAV, Snort, Immundet, SpamCop, SenderBase, Threat Grid, and Talos user communities
- Honeypots
- The Sourcefire Awareness, Education, Guidance, and Intelligence Sharing (AEGIS) program
- Private and public threat feeds
- Dynamic analysis

Cisco Threat Grid

Cisco Threat Grid is a solution that can perform static file analysis, as well as dynamic file analysis (also known as behavioral analysis), by running the files in a controlled and monitored sandbox environment.

- Behavioral analysis is combined with threat intelligence feeds from Talos, as well as with existing security technologies to protect against known and unknown attacks.
- It is also possible to upload suspicious files into a sandbox environment called Glovebox to safely interact with them and observe malware behavior directly.
- Threat Grid is available as an appliance and in the cloud, and it is also integrated into existing Cisco security products and third-party solutions.

Automatic submission of suspicious files and samples is available for products and solutions integrated with Threat Grid. When automatic submission is not available, files can also be uploaded manually into Threat Grid for analysis.

Cisco Advanced Malware Protection (AMP)

Cisco Advanced Malware Protection (AMP) (formerly FireAMP) is a malware analysis and protection solution that goes beyond point-in-time detection.

Cisco AMP provides comprehensive protection for organizations across the full attack continuum:

Attack Time	AMP Processes
Before	Global threat intelligence from Cisco Talos and Cisco Threat Grid feeds into AMP to protect against known and new emerging threats.
During	File reputation to determine whether a file is clean or malicious as well as sandboxing are used to identify threats during an attack.
After	Cisco AMP provides retrospection, indicators of compromise (IoCs), breach detection, tracking, analysis, and surgical remediation after an attack, when advanced malware has slipped past other defenses.

Cisco AMP Components

The architecture of AMP can be broken down into the following components:

- AMP Cloud (private or public)
- AMP connectors
 - AMP for Endpoints (Microsoft Windows, macOS X, Google Android, Apple iOS, and Linux)
 - AMP for Networks (NGFW, NGIPS, ISRs)
 - AMP for Email (ESA)
 - AMP for Web (WSA)
 - AMP for Meraki MX
- Threat intelligence from Cisco Talos and Cisco Threat Grid

Cisco AMP for Endpoints on Apple iOS is known as the Cisco Security Connector (CSC). The CSC incorporates AMP for Endpoints and Cisco Umbrella.

Cisco AMP Components (Cont.)

Figure 25-3 illustrates how all the AMP components come together to form the AMP architecture.

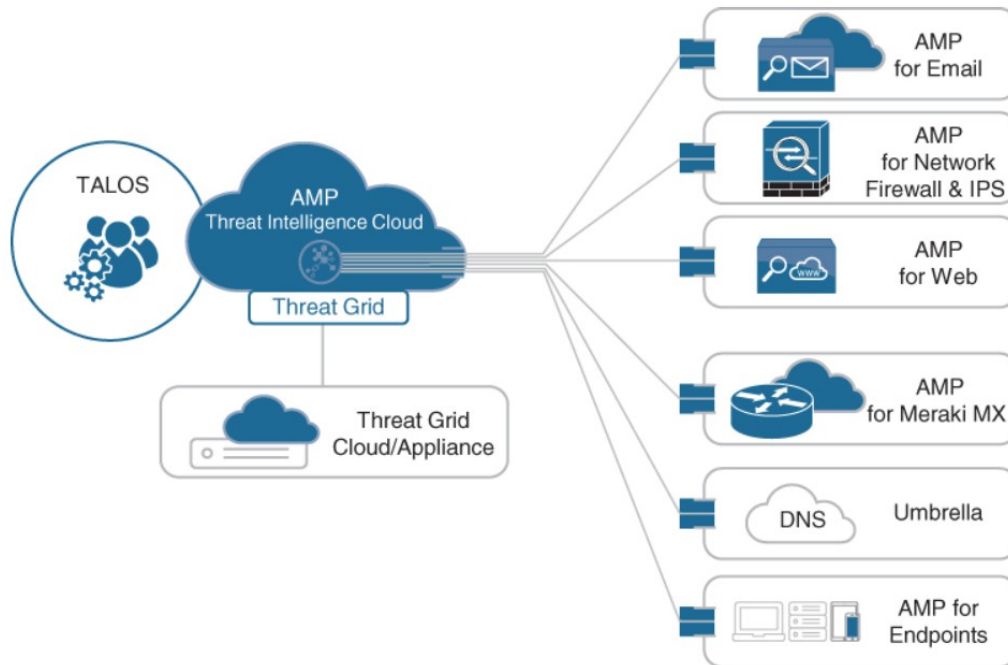


Figure 25-3 AMP Components

Cisco AnyConnect

Cisco AnyConnect Secure Mobility Client is a modular endpoint software product that is *not only* a **VPN client** that provides VPN access through Transport Layer Security (TLS)/Secure Sockets Layer (SSL) and IPsec IKEv2 but also offers enhanced security through various built-in modules, such as a VPN Posture (HostScan) module and an ISE Posture module.

Cisco AnyConnect also includes web security through Cisco Cloud Web Security, network visibility into endpoint flows within Stealthwatch, and roaming protection with Cisco Umbrella.

AnyConnect is supported across the following platforms:

Windows, macOS, iOS, Linux, Android, Windows Phone/Mobile, BlackBerry, and ChromeOS.

TLS/SSL is often used to indicate that either protocol is being discussed. The SSL protocol has been deprecated by the IETF in favor of the more secure TLS protocol, so TLS/SSL can be interpreted as referring to TLS only.

Cisco proprietary

Cisco Umbrella

Cisco Umbrella (formerly known as OpenDNS) provides the first line of defense against threats on the internet by blocking requests to malicious internet destinations (domains, IPs, URLs) using the Domain Name System (DNS) before an IP connection is established or a file is downloaded.

- It is 100% cloud delivered, with no hardware to install or software to maintain.
- The Umbrella global network includes 30 data centers around the world using Anycast DNS, which allows it to guarantee 100% uptime.



Figure 25-4 *Cisco Umbrella Blocking Phishing Website*

Cisco Web Security Appliance (WSA)

The Cisco Web Security Appliance (WSA) is an all-in-one web gateway that includes a wide variety of protections that can block hidden malware from both suspicious and legitimate websites.

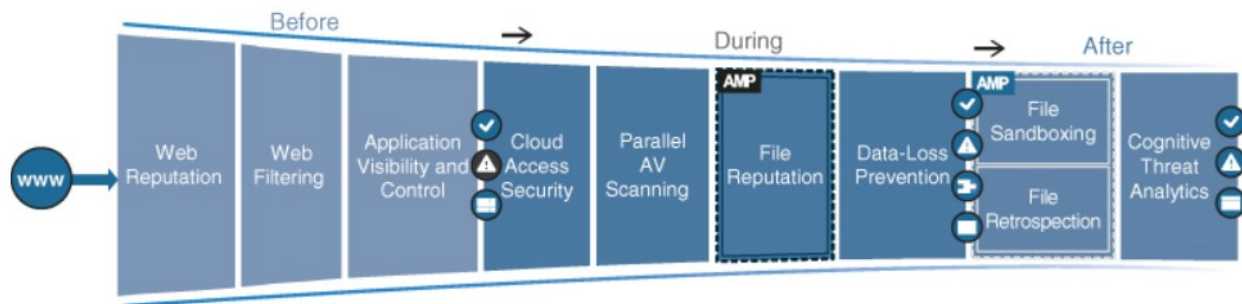


Figure 25-5 WSA Capabilities Across the Attack Continuum

Before	During	After
<ul style="list-style-type: none"> Web reputation filters Web filtering Cisco Application Visibility and Control (AVC) 	<ul style="list-style-type: none"> Cloud access security Parallel antivirus (AV) scanning Layer 4 traffic monitoring File reputation and analysis with Cisco AMP Data loss prevention (DLP) 	<ul style="list-style-type: none"> Continuously inspects for instances of undetected malware and breaches. Global Threat Analytics (GTA)

Before an Attack

Before an attack, the WSA actively detects and blocks potential threats before they happen by applying web reputation filters and URL filtering and by controlling web application usage:

Terms	Description
Web reputation filters	Cisco WSA detects and correlates threats in real time using Talos. Web reputation filtering prevents client devices from accessing dangerous websites containing malware or phishing links.
Web filtering	Traditional URL filtering is combined with real-time dynamic content analysis. This is used to shut down access to websites known to host malware.
Cisco Application Visibility and Control (AVC)	Cisco AVC identifies and classifies the most relevant and widely used web and mobile applications and more than 150,000 micro-applications.

During an Attack

During an attack, the WSA uses security intelligence from cloud access security broker (CASB) providers, Talos, and AMP for networks to identify and block zero-day threats that managed to infiltrate the network:

Terms	Description
Cloud access security	WSA can protect against hidden threats in cloud apps
Parallel antivirus (AV) scanning	WSA enhances malware defense coverage with multiple anti-malware scanning engines
Layer 4 traffic monitoring	WSA scans all traffic, ports, and protocols to detect and block spyware “phone-home” communications
File reputation and analysis with Cisco AMP	WSA assesses files using the latest threat information from Cisco Talos
Data loss prevention (DLP)	WSA uses Internet Control Adaptation Protocol (ICAP) to integrate with DLP solutions from leading third-party DLP vendors

After an Attack

After an attack, Cisco WSA inspects the network continuously for instances of undetected malware and breaches.

- After an initial detection, using Cisco AMP retrospection capabilities, Cisco WSA continues to scan files over an extended period of time, using the latest threat intelligence from Talos and AMP Thread Grid.
- Alerts are sent when a file disposition changes to provide awareness and visibility into malware that evades initial defenses.

Global Threat Analytics (GTA), formerly Cognitive Threat Analytics (CTA), analyzes web traffic, endpoint data from Cisco AMP for Endpoints, and network data from Cisco Stealthwatch Enterprise. It then identifies malicious activity before it can exfiltrate sensitive data.

WSA can be deployed in the cloud, as a virtual appliance, on-premises, or in a hybrid arrangement. All features are available across any deployment option.

Cisco Email Security Appliance (ESA)

The Cisco Email Security Appliance (ESA) enables users to communicate securely via email and helps organizations combat email security threats with a multilayered approach.

Cisco ESA includes the following advanced threat protection capabilities that allow it to detect, block, and remediate threats across the attack continuum:

- **Global threat intelligence** - It leverages real-time threat intelligence from Talos and AMP.
- **Reputation filtering** - ESA blocks unwanted email with reputation filtering.
- **Spam protection** - ESA uses the Cisco Context Adaptive Scanning Engine (CASE) to block spam emails.
- **Forged email detection** - Forged email detection protects high-value targets such as executives against business email compromise (BEC) attacks.

Cisco Email Security Appliance (ESA) (Cont.)

- **Cisco Advanced Phishing Protection (CAPP)** - CAPP combines Cisco Talos with local email intelligence and advanced machine learning techniques to model trusted email behaviors.
- **Cisco Domain Protection (CDP)** - CDP for external email helps prevent phishing emails from being sent using a customer domains.
- **Malware defense** - ESA protects against malware with Cisco AMP for email.
- **Graymail detection and Safe Unsubscribe** - ESA detects and classifies graymail for an administrator to take action on it if necessary. Graymail consists of marketing, social networking, and bulk messages (that is, mailing list emails).
- **URL-related protection and control** - ESA protects against malicious URLs with URL filtering and scanning of URLs in attachments and shortened URLs.

Cisco Email Security Appliance (ESA) (Cont.)

- **Outbreak filters** - Outbreak filters defend against emerging threats and blended attacks by leveraging Cisco Talos.
- **Web interaction tracking** - ESA generates reports that track the end users who click on URLs that have been rewritten by the outbreak filters. The reports include the following information:
 - Top users who clicked on malicious URLs
 - The top malicious URLs clicked by end users
 - Date and time, rewrite reason, and action taken on the URLs
- **Data security for sensitive content in outgoing emails** - Confidential outbound messages that match one of the more than 100 expert policies included with ESA are automatically protected.

Next-Generation Intrusion Prevention System (NGIPS)

A system that passively monitors and analyzes network traffic for potential network intrusion attacks and logs the intrusion attack data for security analysis is known as an intrusion detection system (IDS). A system that provides IDS functions and also automatically blocks intrusion attacks is known as an intrusion prevention system (IPS).

A next-generation IPS (NGIPS) should include IPS functionality as well as the following capabilities:

- Real-time contextual awareness
- Advanced threat protection
- Intelligent security automation
- Unparalleled performance and scalability
- Application visibility and control (AVC) and URL filtering

Next-Generation Intrusion Prevention System (Cont.)

With the acquisition of Sourcefire in 2013, Cisco added the Firepower NGIPS to its portfolio. Following are some of the most important capabilities included with the Cisco Firepower NGIPS:

Features	Advanced Features
Real-time contextual awareness	Centralized management
Advanced threat protection and remediation	Global threat intelligence from the Cisco Talos
Intelligent security automation	Snort IPS detection engine
Unparalleled performance and scalability	High availability and clustering
AVC	Third-party and open-source ecosystem
URL filtering	Integration with Cisco ISE: Quarantine, Unquarantine, Shutdown

Next-Generation Firewall (NGFW)

A firewall is a network security device that monitors incoming and outgoing network traffic and allows or blocks traffic by performing simple packet filtering and stateful inspection based on ports and protocols.

A next-generation firewall (NGFW) can block threats such as advanced malware and application-layer attacks. A NGFW firewall must include:

- Standard firewall capabilities such as stateful inspection
- An integrated IPS
- Application-level inspection (to block malicious or risky apps)
- The ability to leverage external security intelligence to address evolving security threats

NGFW: Management Options

The following management options are available for NGFWs:

- For FTD or Firepower Services software:
 - Firepower Management Center (FMC)
 - Firepower Device Manager (FDM) for small appliances
- For ASA software:
 - The command-line interface (CLI)
 - Cisco Security Manager (CSM)
 - Adaptive Security Device Manager (ASDM)
 - Cisco Defense Orchestrator

FTD or Firepower Services software CLI configuration is not supported. CLI is only available for initial setup and troubleshooting purposes.

Cisco Firepower Management Center (FMC)

The Cisco FMC is a centralized management platform that aggregates and correlates threat events, contextual information, and network device performance data.

The FMC performs event and policy management for the following Firepower security solutions:

- Cisco Firepower NGFW and NGFWv
- Cisco Firepower NGIPS and NGIPSv
- Cisco Firepower Threat Defense for ISR
- Cisco ASA with Firepower Services
- Cisco Advanced Malware Protection (AMP)

Cisco Stealthwatch

Cisco Stealthwatch is a collector and aggregator of network telemetry data that performs network security analysis and monitoring to automatically detect threats that manage to infiltrate a network as well as the ones that originate from within a network.

There are currently two offerings available for Stealthwatch:

- Stealthwatch Enterprise
- Stealthwatch Cloud

Cisco Stealthwatch Enterprise

Stealthwatch Enterprise provides real-time visibility into activities occurring within the network.

- At the core of Stealthwatch Enterprise are the Flow Rate License, the Flow Collector, Management Console, and Flow Sensor. Optional but recommended components include the following:
 - Cisco Stealthwatch Threat Intelligence
 - Cisco Stealthwatch Endpoint
 - Cisco Stealthwatch Cloud
- Stealthwatch Enterprise offers the following benefits:
 - Real-time threat detection
 - Incident response and forensics
 - Network segmentation
 - Network performance and capacity planning
 - Ability to satisfy regulatory requirements

Cisco Stealthwatch Enterprise (Cont.)

Stealthwatch Enterprise requires the following components:

- **Flow Rate License** - The Flow Rate License is required for the collection, management, and analysis of flow telemetry data and aggregates flows at the Stealthwatch Management Console, as well as to define the volume of flows that can be collected
- **Flow Collector** - The Flow Collector collects and analyzes enterprise telemetry data and other types of flow data.
- **Stealthwatch Management Console (SMC)** - The SMC is the control center that aggregates, organizes, and presents analysis from up to 25 Flow Collectors, Cisco ISE, and other sources.

Optional Stealthwatch Enterprise components include the following:

- Flow Sensor
- UDP Director

Cisco Stealthwatch Cloud

Stealthwatch Cloud provides the visibility and continuous threat detection required to secure the on-premises, hybrid, and multicloud environments.

Cisco Stealthwatch Cloud consists of two primary offerings:

- **Public Cloud Monitoring** - Cisco Stealthwatch Cloud Public Cloud Monitoring provides visibility and threat detection in AWS, GCP, and Microsoft Azure cloud infrastructures. It is a SaaS-based solution that can be deployed easily and quickly.
- **Private Network Monitoring** - Cisco Stealthwatch Cloud Private Network Monitoring provides visibility and threat detection for the on-premises network, delivered from a cloud-based SaaS solution.

Stealthwatch Cloud consumes metadata only. The actual packet payloads are never retained or transferred outside the network.

Cisco Identity Services Engine (ISE)

Cisco Identity Services Engine (ISE) is a security policy management platform that provides highly secure network access control (NAC) to users and devices across wired, wireless, and VPN connections.

Some of the most important features ISE include the following:

ISE Features

Streamlined network visibility	Streamlined device onboarding
Cisco Digital Network Architecture (DNA) Center integration	Internal certificate authority
Centralized secure network access control	Device profiling:
Centralized device access control	Endpoint posture service
Cisco TrustSec	Active Directory support
Guest lifecycle management	Cisco Platform Exchange Grid (pxGrid)

Cisco Identity Services Engine (ISE) Example

Example 25-1 shows the type of contextual information Cisco ISE can share with devices integrated with it through pxGrid.

Example 25-1 Contextual Information from Cisco ISE Session Directory

```
Session={ip=[192.168.1.2]
Audit Session Id=0A000001000000120001C0AC
UserName=dewey.hyde@corelab.com
ADUserDNSDomain=corelab.com
ADUserNetBIOSName=corelab,
ADUserResolvedIdentities=dewey.hyde@corelab.com
ADUserResolvedDNs=CN=Dewey Hyde
CN=Users
DC=corelab
DC=com
MacAddresses=[00:0C:C1:31:54:69]
State=STARTED
ANCstatus=ANC_Quarantine
SecurityGroup=Quarantined_Systems
EndpointProfile=VMWare-Device

EndpointProfile=VMWare-Device
NAS IP=192.168.1.1
NAS Port=GigabitEthernet0/0/1
RADIUSAVPairs=[ Acct-Session-Id=0000002F]
Posture Status=null
Posture Timestamp=
LastUpdateTime=Sat Aug 21 11:49:50 CST 2019
Session attributeName=Authorization_Profiles
Session attributeValue=Quarantined_Systems
Providers=[None]
EndpointCheckResult=none
IdentitySourceFirstPort=0
IdentitySourcePortStart=0
```

Network Access Control (NAC)

This section describes multiple network access control (NAC) technologies, such as 802.1x, MAC Authentication Bypass (MAB), and Web Authentication (WebAuth), as well as nextgeneration NAC technologies such as TrustSec and MACsec.

802.1x

IEEE 802.1x (referred to as Dot1x) is a standard for port-based network access control (PNAC) that provides an authentication mechanism for local area networks (LANs) and wireless local area networks (WLANs).

802.1x comprises the following components:

- **Extensible Authentication Protocol (EAP)** - This message format and framework defined by RFC provides an encapsulated transport for authentication parameters.
- **EAP method (also referred to as EAP type)** - Different authentication methods can be used with EAP.
- **EAP over LAN (EAPoL)** - This Layer 2 encapsulation protocol is defined by 802.1x for the transport of EAP messages over IEEE 802 wired and wireless networks.
- **RADIUS protocol** - This is the AAA protocol used by EAP.

802.1x Roles

802.1x network devices have the following roles:

- **Supplicant** - Software on the endpoint communicates and provides identity credentials through EAPoL with the authenticator.
- **Authenticator** - A network access device (NAD) such as a switch or wireless LAN controller (WLC) controls access to the network based on the authentication status of the user or endpoint.
- **Authentication server** - RADIUS server performs authentication of the client.

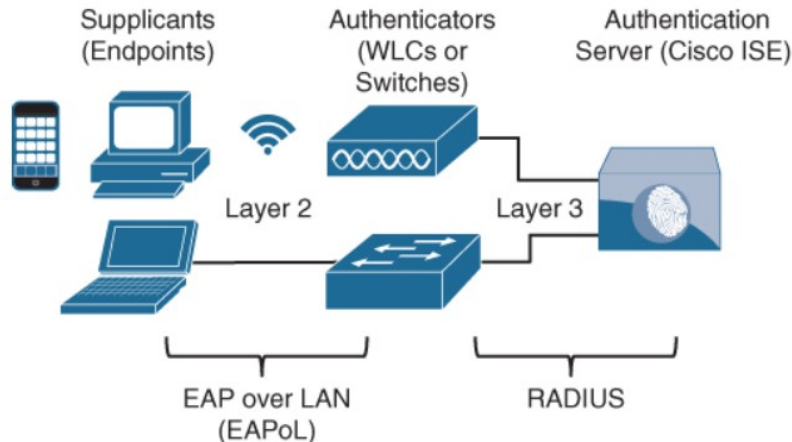


Figure 25-6 802.1x Roles and Components

802.1x Authentication

The EAP identity exchange and authentication occur between the supplicant and the authentication server.

Step 1. When the authenticator notices a port coming up, it starts the authentication process by sending periodic EAP-request/identify frames.

Step 2. The authenticator relays EAP messages between the supplicant and the authentication server.

Step 3. If authentication is successful, the authentication server returns a RADIUS access-accept message.

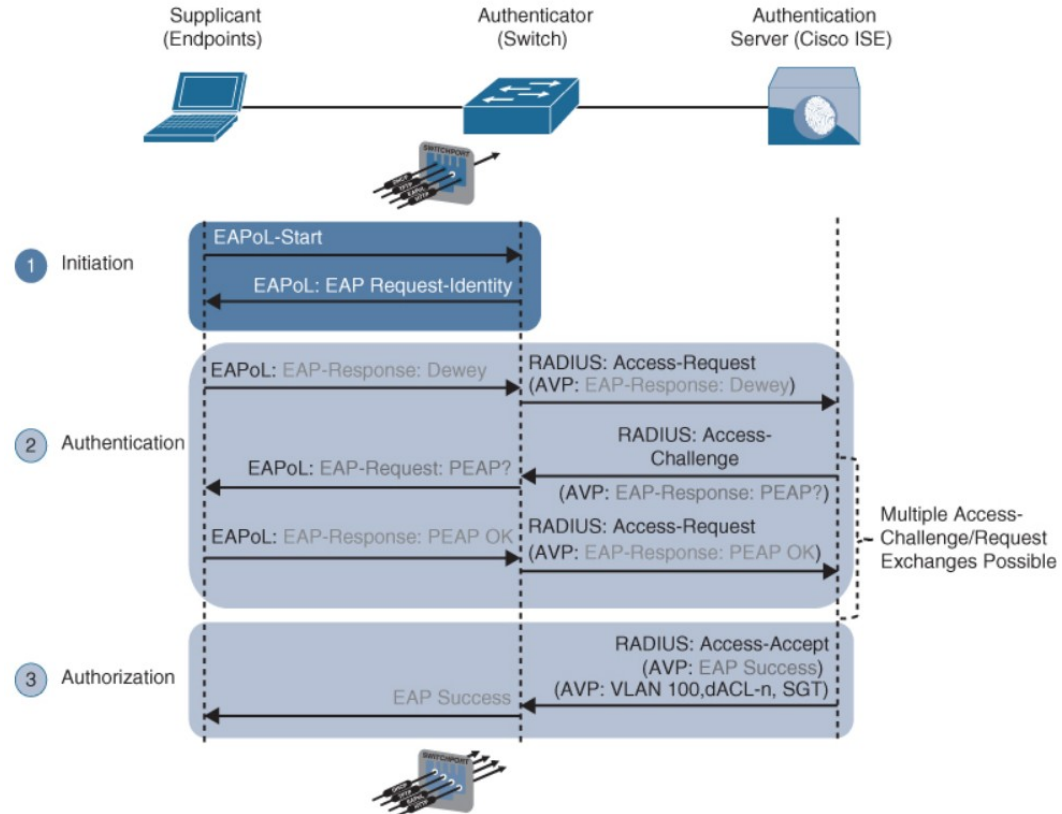


Figure 25-7 Successful 802.1x Authentication Process Flow

EAP Methods

There are many different EAP authentication methods available, most of them based on Transport Layer Security (TLS). The following are the most commonly used EAP methods, which are described in this section:

- EAP challenge-based authentication method
 - Extensible Authentication Protocol-Message Digest 5 (EAP-MD5)
- EAP TLS authentication method
 - Extensible Authentication Protocol-Transport Layer Security (EAP-TLS)
- EAP tunneled TLS authentication methods
 - Extensible Authentication Protocol Flexible Authentication via Secure Tunneling (EAP-FAST)
 - Extensible Authentication Protocol Tunneled Transport Layer Security (EAP-TTLS)
 - Protected Extensible Authentication Protocol (PEAP)
- EAP inner authentication methods
 - EAP Generic Token Card (EAP-GTC)
 - EAP Microsoft Challenge Handshake Authentication Protocol Version 2 (EAP-MSCHAPv2)
 - EAP TLS.

EAP Methods (Cont.)

Following is a description of each of the EAP authentication methods:

- **EAP-MD5** – This uses the MD5 message-digest algorithm to hide the credentials in a hash.
- **EAP-TLS** – This uses the TLS Public Key Infrastructure (PKI) certificate authentication mechanism to provide mutual authentication of supplicant to authentication server and authentication server to supplicant.
- **PEAP** – In PEAP, only the authentication server requires a certificate. PEAP forms an encrypted TLS tunnel between the supplicant and the authentication server

After the tunnel has been established, PEAP uses one of the following EAP authentication inner methods to authenticate the supplicant through the outer PEAP TLS tunnel:

PEAP Authentication		
EAP-MSCHAPv2 (PEAPv0)	EAP-TLS	EAP-TTLS
EAP-GTC (PEAPv1)	EAP-FAST	

EAP Chaining

EAP-FAST includes the option of EAP chaining:

- Supports machine and user authentication inside a single outer TLS tunnel
- Enables machine and user authentication to be combined into a single overall authentication result
- Allows the assignment of greater privileges or posture assessments to users who connect to the network using corporate managed devices

MAC Authentication Bypass (MAB)

MAC Authentication Bypass (MAB) is an access control technique that enables **port-based access control using the MAC address** of an endpoint.

Step 1. The switch initiates authentication by sending an EAPoL identity request message to the endpoint every 30 seconds by default.

Step 2. The switch begins MAB by opening the port to accept a single packet from which it will learn the source MAC address of the endpoint.

Step 3. The RADIUS server determines whether the device should be granted access to the network

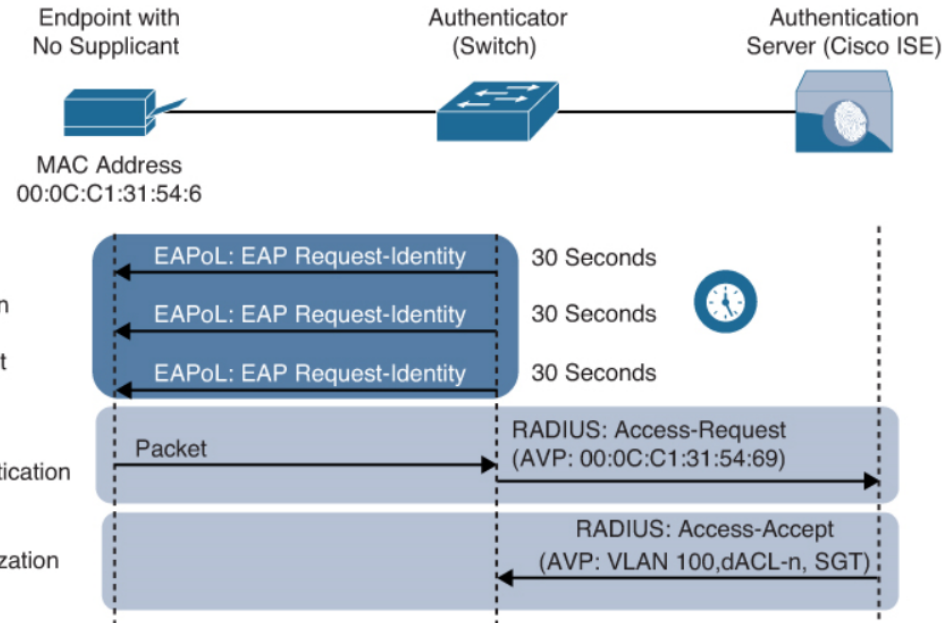


Figure 25-8 Successful MAB Authentication Process Flow © 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential 39

MAC Authentication Bypass (Cont.)

MAC addresses are easily spoofed. For this reason, MAB authenticated endpoints should be given very restricted access and should only be allowed to communicate to the networks and services that the endpoints are required to speak to.

If the authenticator is a **Cisco switch**, then **many authorization options** can be applied as part of the authorization result from the authentication server, including the following:

- Downloadable ACLs (dACLs)
- Dynamic VLAN assignment (dVLAN)
- Security Group Tags (SGT) tags

Web Authentication (WebAuth)

Web Authentication (WebAuth) can be used for endpoints that try to connect to the **network might not have 802.1x** supplicants and **might not know the MAC address** to perform MAB.

- WebAuth, like MAB, can be used as a **fallback** authentication mechanism **for 802.1x**.
- If both MAB and WebAuth are configured as fallbacks for 802.1x, when 802.1x times out a switch **first** attempts to authenticate through **MAB**, and **if it fails**, the switch attempts to authenticate with **WebAuth**.
- Unlike MAB, WebAuth is **only for users** and **not devices** since it requires a web browser and **manual username and password** entry.

There are two types of WebAuth:

1. Local Web Authentication
2. Centralized Web Authentication with Cisco ISE

Local Web Authentication

Local Web Authentication (LWA) is the first form of Web Authentication that was created.

The switch (or wireless controller) **redirects** web traffic (HTTP and/or HTTPS) **to a locally hosted web portal running in the switch** where an end user can enter a username and a password.

- When the switch sends the login credentials on behalf of the user, it is considered to be LWA.
- The LWA web portals are **not customizable**.
- With Cisco switches, there is **no native support for advanced services** such as acceptable use policy (AUP), acceptance pages, password changing capabilities, device registration, and self-registration. For those advanced capabilities, a centralized web portal is required.
- LWA **does not support VLAN assignment**; it supports **only ACL** assignment.
- LWA **doesn't support the change of authorization** (CoA) feature **to apply new policies**. Therefore, access policies cannot be changed based on posture or profiling state, and even administrative changes cannot be made as a result of malware to quarantine the endpoint.

Central Web Authentication with Cisco ISE

Cisco created Centralized Web Authentication (CWA) to overcome LWA's deficiencies.

CWA supports the following:

- CoA for posture profiling, as well as dACL (*downloadable ACLs*) and VLAN authorization options.
- All the advanced services: client provisioning, posture assessments, acceptable use policies, password changing, self-registration, and device registration.

Just like LWA, CWA is **only for endpoints** that have a web browser, where the user can manually enter a username and a password.

With CWA, WebAuth and guest VLAN functions remain mutually exclusive.

Central Web Authentication with Cisco ISE (Cont.)

Authentication for CWA is different from authentication for LWA. The following steps detail how CWA authentication takes place:

Step 1. The endpoint entering the network does not have a configured supplicant or the supplicant is misconfigured.

Step 2. The switch performs MAB, sending the RADIUS access-request to Cisco ISE (the authentication server).

Step 3. The authentication server (ISE) sends the RADIUS result, including a URL redirection, to the centralized portal on the ISE server itself.

Step 4. The endpoint is assigned an IP address, DNS server, and default gateway using DHCP.

Step 5. The end user opens a browser and enters credentials into the centralized web portal.

Step 6. ISE sends a re-authentication change of authorization (CoA-reauth) to the switch.

Step 7. The switch sends a new MAB request with the same session ID to ISE. ISE sends the final authorization result to the switch for the end user.

Enhanced Flexible Authentication (FlexAuth)

By **default**, a Cisco switch configured with 802.1x, MAB, and WebAuth always attempts **802.1x** authentication first, followed by **MAB**, and **finally WebAuth**.

If an endpoint that does **not support 802.1x** tries to connect to the network, it **needs to wait** for a considerable amount of time **before WebAuth** is offered as an authentication option.

- **Enhanced FlexAuth** (also referred to as **Access Session Manager**) addresses this problem by allowing **multiple authentication methods concurrently** (for example, 802.1x and MAB) so that endpoints can be authenticated and brought online more quickly.
- Enhanced FlexAuth is a **key component** of the **Cisco Identity-Based Networking Services (IBNS) 2.0** integrated solution, which offers **authentication, access control, and user policy enforcement**.

Cisco Identity-Based Networking Services (IBNS) 2.0

Cisco IBNS 2.0 is an integrated solution that offers authentication, access control, and user policy enforcement with a common end-to-end access policy that applies to both wired and wireless networks.

It is a combination of the following existing features and products:

- Enhanced FlexAuth (Access Session Manager)
- Cisco Common Classification Policy Language (C3PL)
- Cisco ISE

Cisco TrustSec

TrustSec is a **next-generation access control enforcement solution** developed by Cisco to address the growing operational challenges related to **maintaining firewall rules and ACLs** by using **Security Group Tag (SGT)** tags.

- TrustSec uses SGT tags to perform ingress tagging and egress filtering to enforce access control policy.
- Cisco ISE assigns the SGT tags to users or devices that are successfully authenticated and authorized through 802.1x, MAB, or WebAuth.
- The SGT tag assignment is delivered to the authenticator as an authorization option (in the same way as a dACL). After the SGT tag is assigned, an access enforcement policy (allow or drop) based on the SGT tag can be applied at any egress point of the TrustSec network.
- SGT tags represent the context of the user, device, use case, or function. This means SGT tags are often named after particular roles or business use cases.

SGT tags are referred to as scalable group tags in Cisco Software-Defined Access (SD-Access).

Network Access Control (NAC)

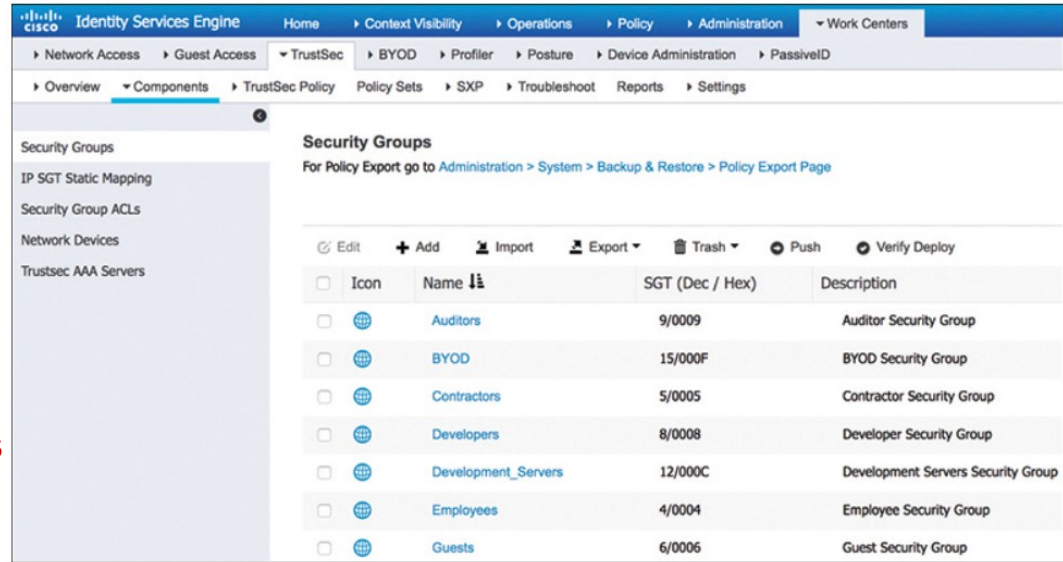
Cisco TrustSec (Cont.)

Figure 25-9 illustrates a list of default SGT tags on Cisco ISE. The SGT tags all have business-relevant names and descriptions.

The SGT name is available on ISE and network devices to create policies; what is actually inserted into a Layer 2 frame SGT tag is a numeric value like the ones shown in the SGT column in decimal and hexadecimal notation.

TrustSec configuration occurs in three phases

- Ingress classification
- Propagation
- Egress enforcement



Icon	Name	SGT (Dec / Hex)	Description
<input type="checkbox"/>	Auditors	9/0009	Auditor Security Group
<input type="checkbox"/>	BYOD	15/000F	BYOD Security Group
<input type="checkbox"/>	Contractors	5/0005	Contractor Security Group
<input type="checkbox"/>	Developers	8/0008	Developer Security Group
<input type="checkbox"/>	Development_Servers	12/000C	Development Servers Security Group
<input type="checkbox"/>	Employees	4/0004	Employee Security Group
<input type="checkbox"/>	Guests	6/0006	Guest Security Group

Figure 25-9 Default SGT Tags in Cisco ISE

Ingress Classification

Ingress classification is the process of assigning SGT tags to users, endpoints, or other resources as they ingress the TrustSec network, and it can happen in one of two ways:

- **Dynamic assignment** - The SGT is assigned dynamically and can be downloaded as an authorization option from ISE when authenticating using 802.1x, MAB, or WebAuth.
- **Static assignment** - In environments such as a data center that do not require 802.1x, MAB, or WebAuth authentication, dynamic SGT assignment is not possible. Static assignment on a device can be one of the following:
 - IP to SGT tag
 - Subnet to SGT tag
 - VLAN to SGT tag
 - Layer 2 interface to SGT tag
 - Layer 3 logical interface to SGT tag
 - Port to SGT tag
 - Port profile to SGT tag

As an alternative to assigning an SGT tag to a port, Cisco ISE added the ability to centrally configure a database of IP addresses and their corresponding SGT tags. Network devices that are SGT capable can download the list from Cisco ISE.

Propagation

Propagation is the process of communicating the mappings to the TrustSec network devices that will enforce policy based on SGT tags.

There are two methods available for propagating an SGT tag—inline tagging (also referred to as native tagging) and the **Cisco-created** protocol **SGT Exchange Protocol (SXP)**:

- **Inline tagging** - With inline tagging, a switch inserts the SGT tag inside a frame to allow upstream devices to read and apply policy.

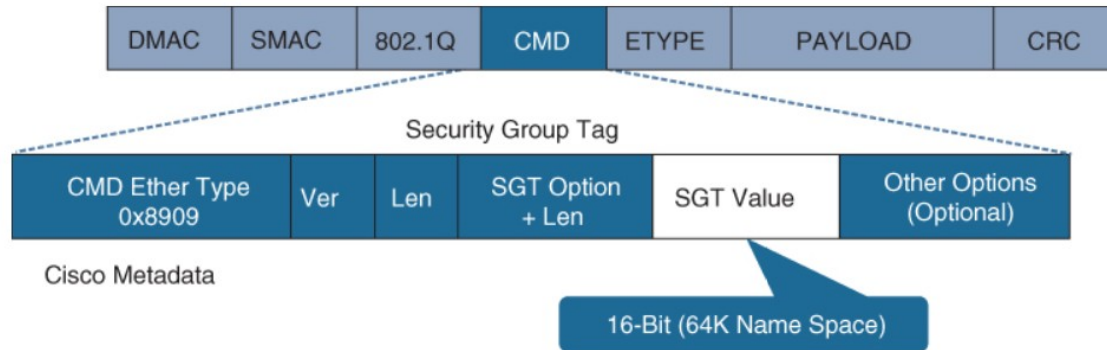


Figure 25-10 Layer 2 Ethernet Frame with an SGT Tag

Propagation (Cont.)

- **SXP propagation** - SXP is a **TCP-based peer-to-peer protocol** used for network devices that do not support SGT inline tagging in hardware.
- Non-inline tagging switches also have an SGT mapping database to check packets against and enforce policy.
- The SXP peer that sends IP-to-SGT bindings is called a **speaker**.
- The IP-to-SGT binding receiver is called a **listener**.
- SXP connections can be single-hop or multi-hop, as shown in Figure 25-11.

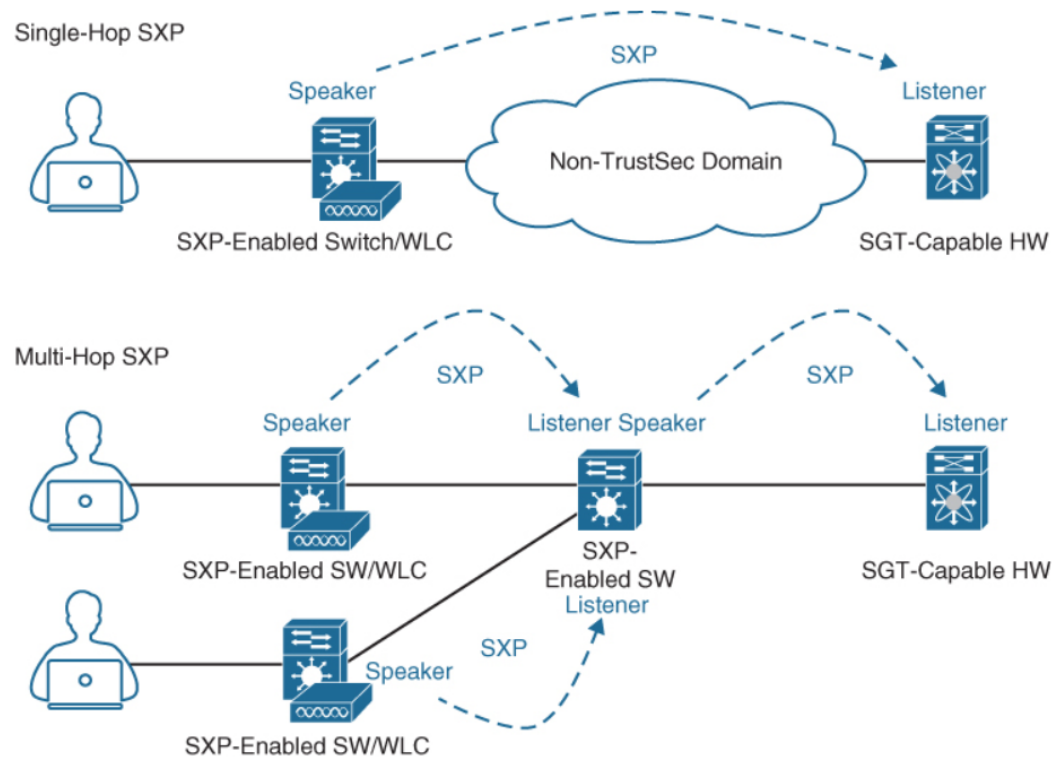


Figure 25-11 Single-Hop and Multi-Hop SXP Connections

Propagation: SPX Example

- Figure 25-12 shows an example of one access switch that supports native tagging. The packets get tagged on the uplink port and through the infrastructure.
- It also shows a switch that is not capable of inline tagging and that uses SXP to update the upstream switch.
- In both cases, the upstream switch continues to tag the traffic throughout the infrastructure.

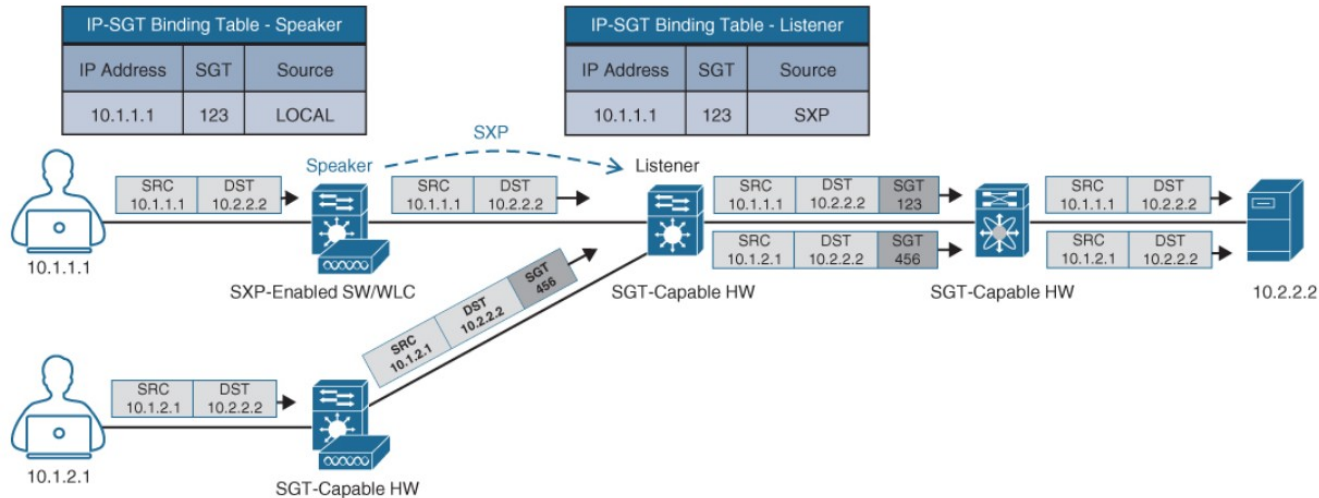


Figure 25-12 *Inline Tagging and SXP Propagation*

Propagation: SPX Peering

Figure 25-13 illustrates an example where a user authenticates to ISE via 802.1x.

The user is connected to a switch that does not support inline tagging or SXP. This means an SGT-to-IP binding cannot be assigned to the user on the switch. The solution is for ISE to assign an SGT to the user by sending a mapping through SXP to an upstream device that supports TrustSec.

Cisco ISE also supports assigning the SGT mapping information to an upstream device through pxGrid.

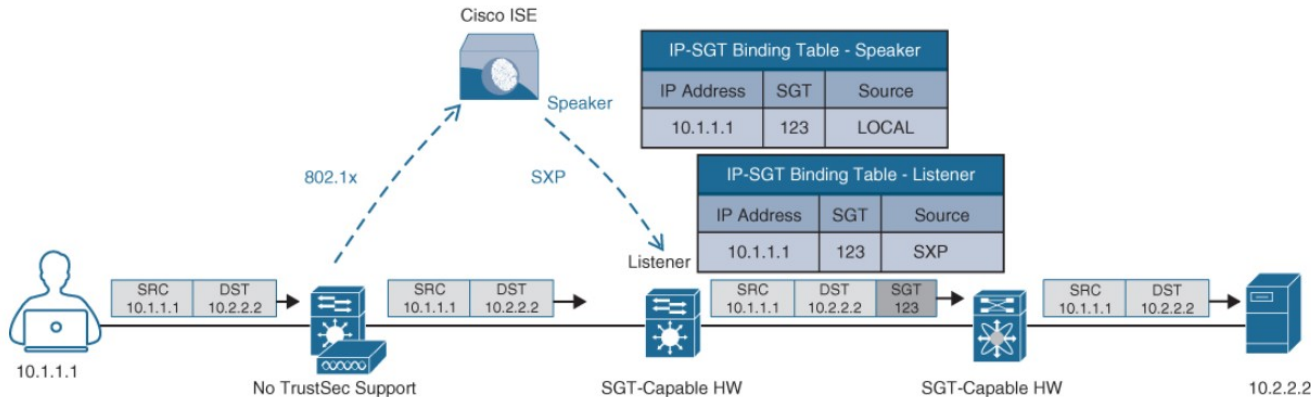


Figure 25-13 SXP Peering Between Cisco ISE and TrustSec-Capable Devices

Egress Enforcement

After the SGT tags have been assigned (classification) and are being transmitted across the network (propagation), policies can be enforced at the egress point of the TrustSec network.

- There are multiple ways to enforce traffic based on the SGT tag, and they can be divided into two major types:
- Security Group ACL (SGACL) - Provides enforcement on routers and switches. Access lists provide filtering based on source and destination SGT tags.
- Security Group Firewall (SGFW) - Provides enforcement on firewalls (such as Cisco ASA and NGFW). Requires tag-based rules to be defined locally on the firewall.

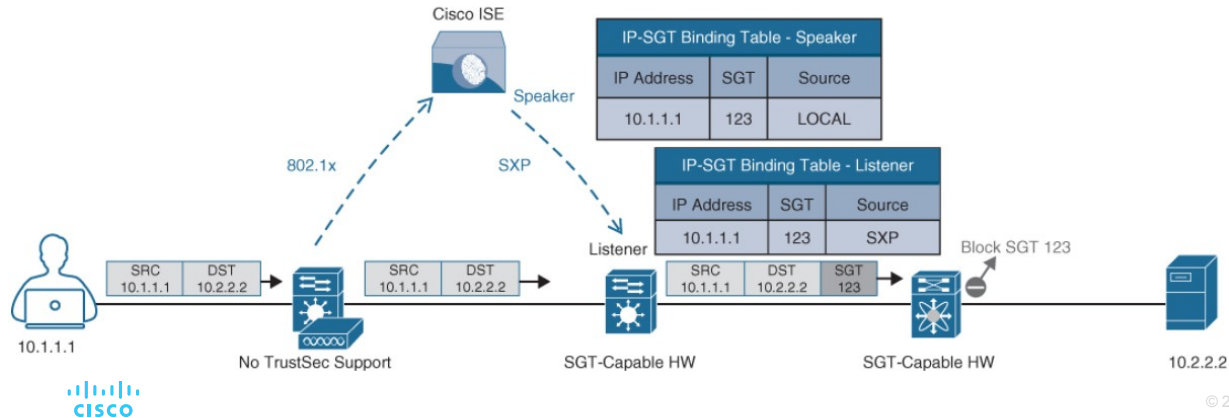


Figure 25-14 *TrustSec Enforcement with SGACL*

Egress Enforcement: SGACL

Figure 25-15 illustrates an SGACL egress policy production matrix from Cisco ISE that allows the defined SGACL enforcements to be visualized.

Figure 25-16 shows the SGACL Permit_FTP configuration on Cisco ISE, which is only allowing FTP traffic (TCP port 21) and denying all other traffic.

Production Matrix Populated cells: 10

Source	Destination: BYOD (15/000F)	Contractors (5/0005)	Development_Ser... (12/000C)	Employees (4/0004)
Auditors (9/0009)	Permit IP		Deny IP	
BYOD (15/000F)	Permit IP		Deny IP	
Contractors (5/0005)		Permit IP	Deny IP	
Developers (8/0008)			Permit IP	
Development_Ser... (12/000C)			Deny IP	
Employees (4/0004)			Deny IP	Permit_FTP

Figure 25-15 SGACL Production Matrix View

Network Access > Guest Access > TrustSec > BYOD > Profiler > Posture > Device Administration

Overview > Components > TrustSec Policy > Policy Sets > SXP > Troubleshoot > Reports > Settings

Security Groups ACLs List > Permit_FTP

Security Group ACLs

- Name: Permit_FTP
- Description: Only allows FTP traffic
- IP Version: IPv4 IPv6 Agnostic
- Security Group ACL content:


```
permit tcp eq 21
deny ip
```

Figure 25-16 Permit FTP SGACL Contents

Egress Enforcement Example

Figure 25-17 illustrates a scenario where only developers have access to the development servers, and any employee trying to access them is blocked.

Traffic is blocked on egress and not on ingress.

This example also illustrates that FTP is the only protocol allowed between employees, while any other type of traffic is blocked. For the employees connected to the same switch, the switch is acting as the ingress and egress point.

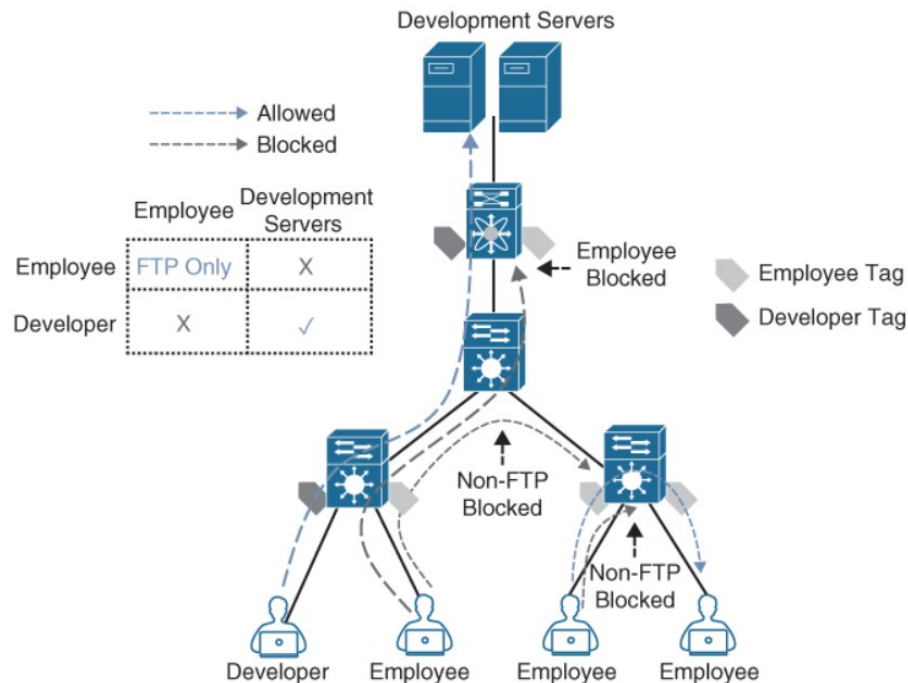


Figure 25-17 SGACL Enforcement Scenario

MACsec

MACsec is an IEEE 802.1AE standards-based Layer 2 hop-by-hop encryption method.

- The traffic is encrypted only on the wire between two MACsec peers and is unencrypted as it is processed internally within the switch. This allows the switch to look into the inner packets for things like SGT tags to perform packet enforcement or QoS prioritization.
- MACsec also leverages onboard ASICs to perform the encryption and decryption rather than having to offload to a crypto engine, as with IPsec.
- MACsec is based on the Ethernet frame format; however, an additional 16-byte MACsec Security Tag field (802.1AE header) and a 16-byte Integrity Check Value (ICV) field are added.
- MACsec provides authentication using Galois Method Authentication Code (GMAC) or authenticated encryption using Galois/Counter Mode Advanced Encryption Standard (AES-GCM).

MACsec Illustrated

Figure 25-18 illustrates the MACsec frame format and how it encrypts the TrustSec SGT tag.

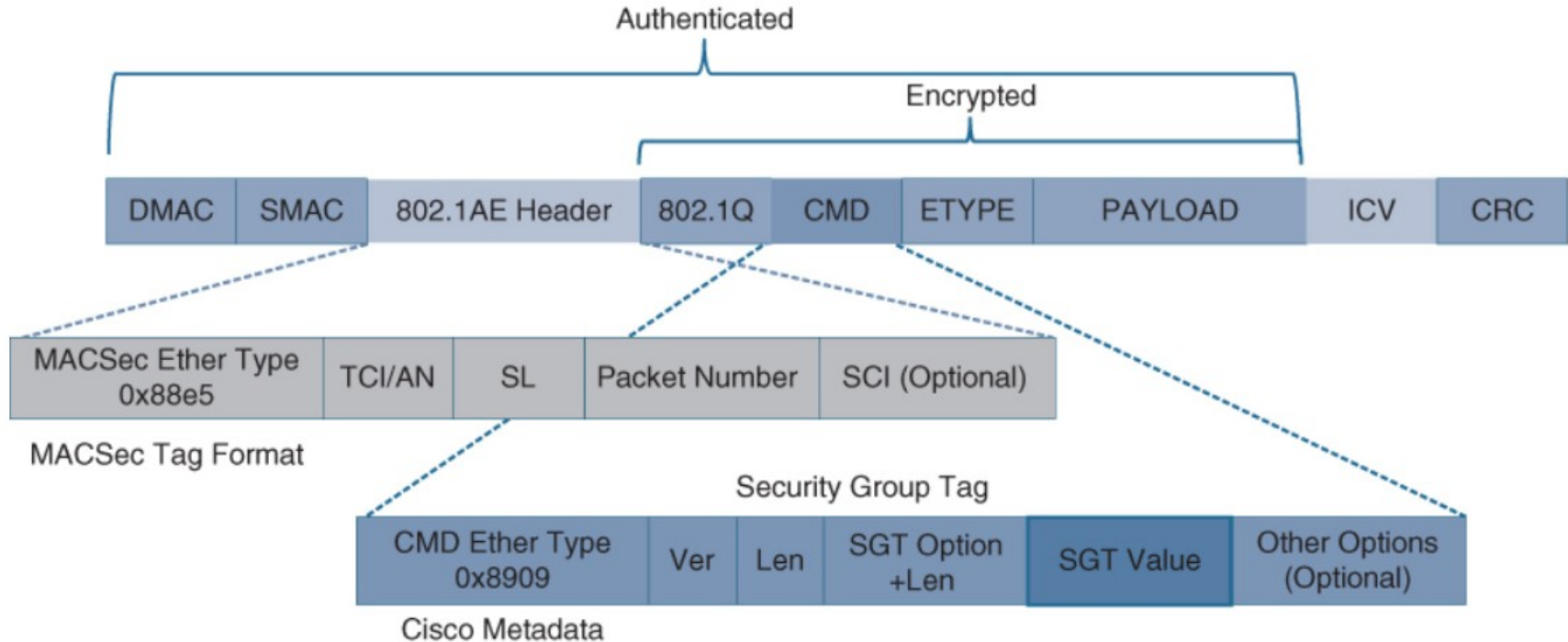


Figure 25-18 *MACsec Ethernet Frame with SGT*

MACsec Tags

The MACsec Security Tag fields are as follows:

- **MACsec EtherType (first two octets)** - Set to 0x88e5, designating the frame as a MACsec frame
- **TCI/AN (third octet)** - Tag Control Information/Association Number field, designating the version number if confidentiality or integrity is used on its own
- **SL (fourth octet)** - Short Length field, designating the length of the encrypted data
- **Packet Number (octets 5–8)** - The packet number for replay protection and building of the initialization vector
- **SCI (octets 9–16)** - Secure Channel Identifier, for classifying the connection to the virtual port

Two MACsec keying mechanisms are available:

- **Security Association Protocol (SAP)** - This is a proprietary Cisco keying protocol used between Cisco switches.
- **MACsec Key Agreement (MKA) protocol** - MKA provides the required session keys and manages the required encryption keys.

Downlink MACsec

Downlink MACsec is the term used to describe the encrypted link between an endpoint and a switch.

- The encryption between the endpoint and the switch is handled by the MKA keying protocol. This requires a MACsec-capable switch and a MACsec-capable supplicant on the endpoint (such as Cisco AnyConnect). The encryption on the endpoint may be handled in hardware or in software, using the main CPU for encryption and decryption.
- The Cisco switch has the ability to force encryption, make encryption optional, or force non-encryption.
- This setting may be configured manually per port (which is not very common) or dynamically as an authorization option from Cisco ISE.
- If ISE returns an encryption policy with the authorization result, the policy issued by ISE overrides anything set using the switch CLI.

Uplink MACsec

Uplink MACsec is the term for encrypting a link between switches with 802.1AE.

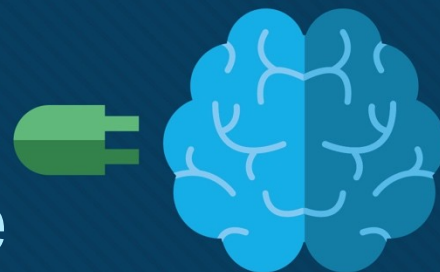
- By default, uplink MACsec uses Cisco proprietary SAP encryption. The encryption is the same AES-GCM-128 encryption used with both uplink and downlink MACsec.
- Uplink MACsec may be achieved manually or dynamically. Dynamic MACsec requires 802.1x authentication between the switches.



Chapter 26: Network Device Access Control and Infrastructure Security

Instructor Materials

CCNP Enterprise: Core Networking



Access Control Lists (ACLs)

- ACLs are sequential lists of access control entries (ACEs) that perform permit or deny packet classification, based on predefined conditional matching statements.
- Packet classification starts at the top (lowest sequence) and proceeds down (higher sequence) until a matching pattern is identified.
- When a match is found, the appropriate action (permit or deny) is taken, and processing stops.
- At the end of every ACL is an implicit deny ACE, which denies all packets that did not match earlier in the ACL.

ACLs

While different kinds of ACLs can be used for packet filtering, only the following types are covered in this chapter:

- **Numbered standard ACLs** - These ACLs define packets based solely on the source network, and they use the numbered entries 1–99 and 1300–1999.
- **Numbered extended ACLs** - These ACLs define packets based on source, destination, protocol, port, or a combination of other packet attributes, and they use the numbered entries 100–199 and 2000–2699.
- **Named ACLs** - These ACLs allow standard and extended ACLs to be given names instead of numbers.
- **Port ACLs (PACLs)** - These ACLs can use standard, extended, named, and named extended MAC ACLs to **filter traffic on Layer 2 switchports**.
- **VLAN ACLs (VACLs)** - These ACLs can use standard, extended, named, and named extended MAC ACLs to **filter traffic on VLANs**.

Wildcard Masks

ACLs use wildcard masks instead of subnet masks to classify packets that are being evaluated.

All that is required to convert a subnet mask into a wildcard mask is to subtract the subnet mask from 255.255.255.255.

$$\begin{array}{r} 255 . 255 . 255 . 255 \\ - 255 . 255 . 128 . 0 \quad \text{Subnet Mask} \\ \hline 0 . 0 . 127 . 255 \quad \text{Wildcard Mask} \end{array}$$

Access Control Lists (ACLs)

Applying ACLs

ACLs have no effect until they are applied to an interface. The next step **after creating an ACL** is to **apply it to an interface**.

In addition to the interface, the **direction** (in or out) in which the ACL needs to be applied must be specified. Cisco routers allow **only one inbound ACL** and **one outbound ACL per interface**.

ACLs can also be used for various other services in addition to applying to interfaces, such as route maps, class maps, NAT, SNMP, virtual terminal (vty) lines, or traffic-classification techniques.

Access Control Lists (ACLs)

Numbered ACLs

The process for defining a numbered **standard** ACL for IOS nodes is as follows:

Step 1. Define the ACL by using the command `access-list number { deny | permit } source [source-wildcard] [log]`. The ACL number can be 1–99 or 1300–1999.

Step 2. Apply the ACL to an interface by using the command `ip access-group {acl-number} {in|out}` under interface configuration mode.

The keywords **any** and **host** can be used as abbreviations for `source [source-wildcard]`.

- Using the keyword **any** is the equivalent to specifying `0.0.0.0 255.255.255.255`, which matches **all packets**.
- The keyword **host** is used to match a **specific host**. It is the equivalent to having specified a **host IP address followed by** a wildcard mask of `0.0.0.0`. The source and source-wildcard reflect a matching pattern for the network prefix that is being matched.

Numbered ACLs (Cont.)

Example 26-1 demonstrates how a numbered standard ACL is created and applied to an interface to deny traffic from the 172.16.0.0/24 subnet and from host 92.168.1.1/32 while allowing all other traffic coming into interface Gi0/1.

Notice that the last ACE in the ACL explicitly permits all traffic (permit any). If this ACE is not included, all traffic will be dropped because of the implicit deny (deny any) at the end of every ACL.

Example 26-1 *Creating and Applying a Numbered Standard ACL*

```
R1(config)# access-list 1 deny 172.16.0.0 0.0.255.255
R1(config)# access-list 1 deny host 192.168.1.1
R1(config)# access-list 1 permit any
R1(config)# interface GigabitEthernet0/1
R1(config-if)# ip access-group 1 in
```

ACE Entry	Networks
Permit any	Permits all networks
permit 172.16.0.0 0.0.255.255	Permits all networks in the 172.16.0.0/16 range
permit host 192.168.1.1	Permits only the 192.168.1.1/32 network

Numbered Extended ACLs

The process for defining a numbered **extended** ACL is as follows:

Step 1. Define the ACL by using the command `access-list acl-number {deny|permit} protocol source source-wildcard destination destination-wildcard [protocol-options] [log | log-input]`. The ACL number can be 100–199 or 2000–2699.

Step 2. Apply the ACL to an interface by using the command `ip access-group {acl-number} {in|out}` under interface configuration mode.

As with standard ACLs, source `source-wildcard` and destination `destination-wildcard` can be defined to match a single host with the `host` keyword or match any subnet with the `any` keyword.

The `protocol-options` keyword differs based on the protocol specified in the ACE.

Numbered Extended ACLs (Cont.)

Example 26-2 demonstrates how a numbered extended ACL is created and applied to an interface to **block all Telnet and ICMP traffic** as well as **deny all IP traffic from host 10.1.2.2 to host 10.1.2.1**. Notice how Telnet's TCP port 23 is being matched with the `eq` keyword.

Example 26-2 *Creating and Applying Numbered Extended ACLs*

```
R1(config)# access-list 100 deny tcp any any eq 23
R1(config)# access-list 100 deny icmp any any
R1(config)# access-list 100 deny ip host 10.1.2.2 host 10.1.2.1
R1(config)# access-list 100 permit ip any any
R1(config)# interface GigabitEthernet0/1
R1(config-if)# ip access-group 100 in
```

Access Control Lists (ACLs)

Named ACLs

Named ACLs allow for ACLs to be named, which makes administering ACLs much easier as long as proper ACL naming conventions are followed. To create and apply a named ACL, follow these steps:

Step 1. Define the ACL by using the command `ip access-list standard|extended {acl-number | acl-name}`. Entering this command places the CLI in ACL configuration mode.

Step 2. Configure the specific ACE in ACL configuration mode by using the command `[sequence] {permit | deny} source source-wildcard`.

Step 3. Apply the ACL to an interface by using the command `ip access-group {acl-number | acl-name} {in|out}` under interface configuration mode.

Standard and Extended ACLs

Example 26-3 shows how **named and numbered standard** and **extended** ACLs are created and applied to an interface.

Example 26-3 *Standard and Extended Named ACLs*

Named Standard ACL

```
R1(config)# ip access-list standard STANDARD_ACL
R1(config-std-nacl)# deny 172.16.0.0 0.0.255.255
R1(config-std-nacl)# deny host 192.168.1.1
R1(config-ext-nacl)# permit any
R1(config-ext-nacl)# exit
R1(config)# interface GigabitEthernet0/1
R1(config-if)# ip access-group STANDARD_ACL in
```

Numbered Standard ACL

```
R1(config)# access-list 1 deny 172.16.0.0 0.0.255.255
R1(config)# access-list 1 deny host 192.168.1.1
R1(config)# access-list 1 permit any
R1(config)# interface GigabitEthernet0/1
R1(config-if)# ip access-group 1 in
```

Named Extended ACL

```
R1(config)# ip access-list extended EXTENDED_ACL
R1(config-ext-nacl)# deny tcp any any eq 23
R1(config-ext-nacl)# deny icmp any any
R1(config-ext-nacl)# deny ip host 10.1.2.2 host 10.1.2.1
R1(config-ext-nacl)# permit ip any any
R1(config-ext-nacl)# exit
R1(config)# interface GigabitEthernet0/1
R1(config-if)# ip access-group EXTENDED_ACL in
```

Numbered Extended ACL

```
R1(config)# access-list 100 deny tcp any any eq 23
R1(config)# access-list 100 deny icmp any any
R1(config)# access-list 100 deny ip host 10.1.2.2 host 10.1.2.1
R1(config)# access-list 100 permit ip any any
R1(config)# interface GigabitEthernet0/1
R1(config-if)# ip access-group 100 in
```


Access Control Lists (ACLs)

Port ACLs (PACLs)

Access lists applied on Layer 2 ports are called port access control lists (PACLs). PACLs can be **standard**, **extended**, or **named IPv4 ACLs for Layer 3**, and **they can be named MAC address ACLs for Layer 2**. PACLs have a **few restrictions** that vary from platform to platform. The following are some of the most common restrictions:

- PACLs **only** support filtering **incoming traffic** on an interface (no outbound filtering support).
- PACLs **cannot filter Layer 2 control frames**, such as CDP, VTP, DTP, PAgP, UDLD, and STP.
- PACLs are **supported only in hardware**.
- PACLs **do not support** ACLs to **filter IPv6, ARP, or Multiprotocol Label Switching (MPLS)** traffic.
 - A Layer 2 port is a physical LAN or trunk port that belongs to a VLAN.
 - PACL takes effect and overwrites the effect of other ACLs (IOS ACL, VACL)

Access Control Lists (ACLs)

Applying a PACL

An IPv4 PACL is applied to an interface with the `ip access-group access-list in` command.

Example 26-4 shows a PACL applied to a Layer 2 interface Gi0/1 to block ICMP, Telnet traffic, and host 10.1.2.2 access to host 10.1.2.1.

Example 26-4 *Applying a PACL*

```
R1(config)# ip access-list extended PACL
R1(config-ext-nacl)# deny tcp any any eq 23
R1(config-ext-nacl)# deny icmp any any
R1(config-ext-nacl)# deny ip host 10.1.2.2 host 10.1.2.1
R1(config-ext-nacl)# permit ip any any
R1(config-ext-nacl)# exit
R1(config)# interface GigabitEthernet0/1
R1(config-if)# switchport
R1(config-if)# ip access-group PACL in
```

Access Control Lists (ACLs)

VLAN ACLs (VACLs)

Access lists applied to VLANs are called VLAN access control lists (VACLs). VACLs can filter traffic that is bridged within a VLAN or that is routed into or out of a VLAN. To create and apply a VACL, follow these steps:

Step 1. Define a **VLAN access map** by using the command `vlan access-map name sequence`.

Step 2. Configure the **match statement** by using the command `match { ip address { acl-number | acl-name } | mac address acl-name }`.

Step 3. Configure the **action statement** by using the command `action forward|drop [log]`. The action statement specifies the action to be taken when a match occurs.

Step 4. Apply the VACL by using the command `vlan filter vlan-access-map-name vlan-list`. **vlan-list** can be a **single VLAN**, a **range** of VLANs (such as 5–30), or a comma-separated **list** of multiple VLANs (such as 1,2–4,6)

Creating and Applying a VACL

Example 26-5 shows a VLAN access map applied to VLAN 20 to drop ICMP and Telnet traffic and allow other traffic.

Notice that the named ACLs, ICMP and TELNET, **only include** ACEs with a **permit statement**.

This is **because**

- the **ACLs** are **only** used **as matching criteria** by the **VLAN access maps**, **while**
- the **VLAN access maps** are configured with the **action to drop the matched traffic**.

Example 26-5 Creating and Applying a VACL

```
SW1(config)# ip access-list extended ICMP
SW1(config-ext-nacl)# permit icmp any any
SW1(config-ext-nacl)# exit
```

```
SW1(config)# ip access-list extended TELNET
SW1(config-ext-nacl)# permit tcp any any eq 23
SW1(config-ext-nacl)# exit
```

```
SW1(config)# ip access-list extended OTHER
SW1(config-ext-nacl)# permit ip any any
SW1(config-ext-nacl)# exit
```

```
SW1(config)# vlan access-map VACL 20 10
SW1(config-access-map)# match ip address ICMP
SW1(config-access-map)# action drop
SW1(config-access-map)# exit
```

```
SW1(config)# vlan access-map VACL 20 20
SW1(config-access-map)# match ip address TELNET
SW1(config-access-map)# action drop log
SW1(config-access-map)# exit
```

```
SW1(config)# vlan access-map VACL 20 30
SW1(config-access-map)# match ip address OTHER
SW1(config-access-map)# action forward
```

```
SW1(config)# vlan filter VACL_20 vlan-list 20
```

PACL, VACL, and RACL Interaction

When a PACL, a VACL, and a RACL (Router based ACL) are all configured in the same VLAN, the ACLs are applied in a **specific order**, depending on whether the incoming traffic needs to be **bridged** or **routed**.

Bridged traffic processing order (within the same VLAN):

1. Inbound PACL on the switchport (for example, VLAN 10)
2. Inbound VACL on the VLAN (for example, VLAN 10)
3. Outbound VACL on the VLAN (for example, VLAN 10)

Routed traffic processing order (across VLANs):

1. Inbound PACL on the switchport (for example, VLAN 10)
2. Inbound VACL on the VLAN (for example, VLAN 10)
3. Inbound ACL on the SVI (for example, SVI 10)
4. Outbound ACL on the SVI (for example, SVI 20)
5. Outbound VACL on the VLAN (for example, VLAN 20)

Terminal Lines and Password Protection

- Password protection to control or restrict access to the CLI to protect the router from unauthorized remote access and unauthorized local access is the most common type of security that needs to be implemented.

Terminal Lines

There are three basic methods to gain access to the CLI of an IOS device:

- **Console port (cty) line** - On any IOS device, this appears in configuration as line con 0 and in the output of the command show line as cty. The console port is mainly used for local system access using a console terminal.
- **Auxiliary port (aux) line** - This appears in the configuration as line aux 0. The aux port is mainly used for remote access into the device through a modem.
- **Virtual terminal (vty) lines** - These lines are displayed by default in the configuration as line vty 0 4. They are used solely for remote Telnet and SSH connections. They are virtual because they are logical lines with no physical interface associated to them.

Terminal Lines and Password Protection

Password Protection

Each of these types of terminal lines should be password protected. There are three ways to add password protection to the lines:

- Using a **password** configured **directly on the line** (**not recommended**)
- Using **username-based** authentication (recommended as a **fallback**)
- Using an **AAA server**: Highly recommended and covered later in this chapter, in the section “Authentication, Authorization, and Accounting (AAA)”

Password Types

There are five available password types in Cisco IOS:

- **Type 0 passwords** - These passwords are the **most insecure** because they are **not encrypted** and are **visible** in the device configuration in plaintext. The command `enable password` is an example of a command that uses a type 0 password.
- **Type 5 passwords** - These passwords use an improved Cisco proprietary encryption algorithm that makes use of the **MD5 hashing algorithm**. The command `enable secret` specifies an additional layer of security over the command `enable password`.
- **Type 7 passwords** - These passwords use a **Cisco proprietary** Vigenere cypher encryption algorithm and are known to be **weak**. Type 7 encryption is **enabled by** the command `service password-encryption`.
- **Type 8 passwords** - Type 8 passwords specify a Password-Based Key Derivation Function 2 (PBKDF2) with a SHA-256 hashed secret and are **considered to be uncrackable**.
- **Type 9 passwords** - These use the SCRYPT hashing algorithm. Just like type 8 passwords, they are **considered to be uncrackable**.

Password Encryption

The **service password-encryption** command in global configuration mode is used to encrypt type 0 passwords in the configuration (for example, BGP passwords) or over a plaintext session such as Telnet in an effort to prevent unauthorized users from viewing the password.

???

Passwords configured prior to configuring the command **service password-encryption** are not encrypted and must be reentered into the configuration. Password encryption is applied to all type 0 passwords, including authentication key passwords; cty, aux, and vty line passwords; and BGP neighbor passwords.

Unfortunately, the command **service password-encryption** encrypts passwords with type 7 encryption, which is easily reversible.

Username and Password Authentication

Username accounts can be used for several applications, such as console, aux, and vty lines. To establish a username and password login authentication system, you can create usernames on a device for all device users or groups. There are three different ways to configure a username on IOS:

- Using the command **username** *{username}* **password** *{password}* configures a **plaintext** password (**type 0**).
- Using the command **username** *{username}* **secret** *{password}* provides **type 5** encryption.
- Using the command **username** *{username}* **algorithm-type** *{md5 | sha256 | scrypt}* **secret** *{password}* provides **type 5, type 8, or type 9** encryption, respectively.

The third command is recommended because it allows for the highest level of password encryption (type 8 and type 9).

Configuring Line Local Password Authentication

To enable password authentication on a line, the following two commands are required under line configuration mode:

- **password** *password* to configure the password
- **login** to enable password checking at login

In Example 26-7, a password is configured for all users attempting to connect to the `cty`, `vty`, and `aux` lines.

Example 26-7 *vty, cty, and aux Lines with Password-Based Authentication*

```
R1# show running-config | section line
Building configuration...

line con 0
password My.C0n5ole.Pes5
login
line aux 0
password My.AuX.Pes5
login
line vty 0 4
password My.vTy.Pes5
login
!
end
```

Verifying Line Local Password Authentication

Example 26-8 shows an example in which the console line password is being tested.

All that is required to test the password is to log off the console and log back in again using the configured console password.

Example 26-8 *Console Password Test*

```
R1# exit
```

```
Router con0 is now available
```

```
Press RETURN to get started.
```

```
User Access Verification
```

```
Password:
```

```
! Password entered here is not displayed by the router
```

```
Router>
```

Configuring Line Local Username & Password Authentication

To enable username and password authentication, the following two commands are required:

- The command **username** in global configuration mode (using one of the options shown in the “Username and Password Authentication” section, earlier in this chapter).
- The command **login local** under line configuration mode to enable username-based authentication at login.

Example 26-9 shows three usernames (type0, type5, and type9) with different password encryptions each that are allowed to log in to the device

Example 26-9 Local Username-Based Authentication for a vty Line

```
R1# show running-config
Building configuration...

!
! Output Omitted for Brevity

username type0 password 0 weak

username type5 secret 5 $1$b1Ju$kZbBS1Pyh4QzwXyZ1kSZ2/

username type9 secret 9 $9$vPpMf8elb4RVV8$seZ/bDAx1uV4yH75Z/nwUuegLJDVCCc4UXOAE8JGsa0

!
! Output Omitted for Brevity

line con 0
login local

line aux 0
login local

line vty 0 4
login local

!
end
```

Verifying Line Local Username & Password Authentication

Example 26-10 shows user type5 establishing a Telnet session from R2 into R1 using username-based authentication.

Example 26-10 *Verifying Local Username-Based Authentication for vty Lines*

```
! Telnet session initiated from R2 into R1

R2# telnet 10.1.12.1
Trying 10.1.12.1 ... Open

User Access Verification

Username: type5
Password:

! Password entered is not displayed by the router

R1>
```

Privilege Levels & Role-Based Access Control (RBAC)

The Cisco IOS CLI by **default** includes **three privilege levels**, each of which defines what commands are available to a user:

- **Privilege level 0** - Includes the **disable**, **enable**, **exit**, **help**, and **logout** commands.
- **Privilege level 1:** - Also known as **User EXEC mode**. The command **prompt** in this mode includes a **greater-than sign** (R1>). From this mode it is not possible to make configuration changes; in other words, the command **configure terminal** is not available.
- **Privilege level 15** - Also known as **Privileged EXEC mode**. This is the highest privilege level, where **all CLI commands are available**. The command **prompt** in this mode includes a **hash sign** (R1#).

The global configuration command **privilege** *{mode}* **level** *{level}* *{command string}* is used to change or set a privilege level for a command to any of these levels.

Configuring a Username with Privilege Level

Example 26-11 shows a configuration where the user **noc** is created along with the **type 9** (scrypt) secret password `cisco123`.

Notice that the privilege level is also configured to **level 5** as part of the username command.

Level 5 allows the user to go into any interface on the router and shut it down, unshut it, and configure an IP address on it.

Example 26-11 *Configuring a Username with Privilege Level*

```
R1(config)# username noc privilege 5 algorithm-type scrypt secret cisco123
R1(config)# privilege exec level 5 configure terminal
R1(config)# privilege configure level 5 interface
R1(config)# privilege interface level 5 shutdown
R1(config)# privilege interface level 5 no shutdown
R1(config)# privilege interface level 5 ip address
```

Terminal Lines and Password Protection

Verifying Privilege Levels

The example shows a quick test to verify that the only commands allowed for privilege level 5 users are those specified by the privilege level command.

```
R1# telnet 1.2.3.4
Trying 1.2.3.4 ... Open

User Access Verification

Username: noc
Password: cisco123
```

```
R1# show privilege
```

```
Current privilege level is 5
```

```
R1#
```

```
R1# configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
R1(config)# interface gigabitEthernet 0/1
```

```
R1(config-if)# ?
```

```
Interface configuration commands:
```

```
default  Set a command to its defaults
exit     Exit from interface configuration mode
help     Description of the interactive help system
ip       Interface Internet Protocol config commands
no       Negate a command or set its defaults
shutdown Shutdown the selected interface
```

```
R1 (config-if)# ip ?
```

```
Interface IP configuration subcommands:
```

```
address Set the IP address of an interface
```

```
R1(config-if)# ip address 10.1.1.1 255.255.255.0
```

```
R1(config-if)# no ?
```

```
ip       Interface Internet Protocol config commands
shutdown Shutdown the selected interface
```

```
R1(config-if)# no shutdown
```

```
R1(config-if)#
```

```
*Apr 27 18:14:23.749: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
```

```
*Apr 27 18:14:24.750: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1,
changed state to up
```

```
R1(config-if)#
```

```
R1(config-if)# shutdown
```

```
R1(config-if)# end
```

```
*Apr 27 18:14:38.336: %LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state
to administratively down
```

```
*Apr 27 18:14:39.336: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1,
changed state to down
```

```
R1#
```

```
*Apr 27 18:14:40.043: %SYS-5-CONFIG_I: Configured from console by noc on vty0
(1.2.3.4)
```

```
R1#
```

Controlling Access to vty Lines with ACLs

Access to the vty lines of an IOS device can be further secured by applying inbound ACLs on them, allowing access only from a restricted set of IP addresses. Outbound vty connections from an IOS device can also be controlled by applying outbound ACLs to vtys.

To apply a standard or an extended access list to a vty line, use the command **access-class** *{access-list-number|access-list-name}* **{in|out}** under line configuration mode. The **in** keyword applies an inbound ACL, and the **out** keyword applies an outbound ACL.

Verifying Access to vty Lines with ACLs

Example 26-13 demonstrates R1 using Telnet to get into R2 before and after applying an ACL to R2's vty line. R1 is configured with IP address 10.12.1.1 and R2 with 10.12.1.2. The ACL being applied to R2's vty line is meant to block vty access into it from R1.

Example 26-13 Verifying Access to vty Lines with ACLs

! Prior to applying an ACL to R2's vty line, R1 is allowed to telnet into R2

```
R1# telnet 10.12.1.2
Trying 10.12.1.2... Open

User Access Verification

Username: noc
Password:

R2#
R2# exit
```

(Connection to 10.12.1.2 closed by foreign host)

! Access list to deny R1's IP address is created and applied to the vty lines 0 to 4

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# access-list 1 deny 10.12.1.1
R2(config)# access-list 1 permit any
R2 (config)# line vty 0 4
R2(config-line)# access-class 1 in
R2(config-line)# end
R2#
R2# show running-config | section line vty
line vty 0 4
  access-class 1 in
  login local
R2#
*Apr 27 19:49:45.599: %SYS-5-CONFIG_I: Configured from console by console
```

After applying an ACL to R2's vty line, R1 is not allowed to telnet into R2

```
R1# telnet 10.12.1.2
Trying 10.12.1.2 ...
! Connection refused by remote host
```

Controlling Access to vty Lines Using Transport Input

Another way to further control what type of protocols are allowed to access the vty lines is to use the command `transport input {all | none | telnet | ssh}` under line configuration mode.

Table 26-3 includes a description for each of the transport input command keywords.

Keyword	Description
All	Allows Telnet and SSH
None	Blocks Telnet and SSH
telnet	Allows Telnet only
ssh	Allows SSH only
telnet ssh	Allows Telnet and SSH

Table 26-3 Transport Input Command Keyword Description

Controlling Access to vty Lines Using Transport Input

Example 26-14 shows the vty lines from 0 to 4 configured with different `transport input` command keywords.

Keep in mind that vty lines are evaluated from the top (vty 0) onward, and each vty line accepts only one user.

Example 26-14 *vty Lines with Different transport input Keywords*

```
line vty 0
  login local
  transport input all
line vty 1
  login local
  transport input none
line vty 2
  login local
  transport input telnet
line vty 3
  login local
  transport input ssh
line vty 4
  login local
  transport input telnet ssh
```

Verifying Access to vty Lines Using Transport Input

Example 26-15 demonstrates how Telnet sessions are assigned to different vty lines on R1.

Example 26-15 Verifying Access to vty Lines

! An asterisk to the left of the row indicates the line is in use
! The output below shows a user is connected into the console (cty)

```
R1# show line
  Tty Typ      Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns  Int
*   0 CTY          - -      - - -    0       0    0/0     -
    1 AUX    9600/9600  - -      - - -    0       0    0/0     -
    578 VTY          - -      - - -    1       0    0/0     -
    579 VTY          - -      - - -    0       0    0/0     -
    580 VTY          - -      - - -    0       0    0/0     -
    581 VTY          - -      - - -    0       0    0/0     -
    582 VTY          - -      - - -    0       0    0/0     -
```

R1#

! Telnet connection from R2 into R1 is established

```
R2# telnet 10.1.12.1
Trying 10.1.12.1 ... Open
```

User Access Verification

Username: noc

Password:

R1>

! The asterisk in the output of show line on R1 indicates the first vty 0 is now in use.
! vty 0 is mapped to vty 578 automatically.

```
R1# show line
```

```
  Tty Typ      Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns  Int
*   0 CTY          - -      - - -    0       0    0/0     -
    1 AUX    9600/9600  - -      - - -    0       0    0/0     -
*  578 VTY          - -      - - -    2       0    0/0     -
    579 VTY          - -      - - -    0       0    0/0     -
    580 VTY          - -      - - -    0       0    0/0     -
    581 VTY          - -      - - -    0       0    0/0     -
    582 VTY          - -      - - -    0       0    0/0     -
```

R1#

Verifying Access to vty Lines Using Transport Input (Cont.)

```
! Telnet connection from R3 into R1 is established
```

```
R3# telnet 10.1.13.1
```

```
Trying 10.1.13.1 ... Open
```

```
User Access Verification
```

```
Username: noc
```

```
Password:
```

```
R1>
```

```
! The output of show line on R1 indicates the vty 0 and vty 2 are now in use
```

```
! vty 2 is mapped to vty 580
```

```
R1# show line
```

Tty	Typ	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	0	0	0/0	-
1	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
* 578	VTY		-	-	-	-	-	2	0	0/0	-
579	VTY		-	-	-	-	-	0	0	0/0	-
* 580	VTY		-	-	-	-	-	1	0	0/0	-
581	VTY		-	-	-	-	-	0	0	0/0	-
582	VTY		-	-	-	-	-	0	0	0/0	-

```
R1#
```

```
! Telnet connection from R4 into R1 is established
```

```
R4# telnet 10.1.14.1
```

```
Trying 10.1.14.1 ... Open
```

```
User Access Verification
```

```
Username: noc
```

```
Password:
```

```
R1>
```

```
! The output of show line on R1 indicates the vty 0, vty 2 and vty 4 are now in use
```

```
! vty 4 is mapped to vty 582. This leaves no more vty lines available for telnet
```

```
R1# show line
```

Tty	Typ	Tx/Rx	A	Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
* 0	CTY		-	-	-	-	-	0	0	0/0	-
1	AUX	9600/9600	-	-	-	-	-	0	0	0/0	-
* 578	VTY		-	-	-	-	-	2	0	0/0	-
579	VTY		-	-	-	-	-	0	0	0/0	-
* 580	VTY		-	-	-	-	-	1	0	0/0	-
581	VTY		-	-	-	-	-	0	0	0/0	-
* 582	VTY		-	-	-	-	-	1	0	0/0	-

```
! Trying to telnet into R1 from R5 will fail since there are no more vtys available for telnet
```

```
R5# telnet 10.1.15.1
```

```
Trying 10.1.15.1 ...
```

```
% Connection refused by remote host
```

```
R5#
```


Enabling SSH vty Access

Telnet session packets are sent in plaintext, and this makes it very easy to sniff and capture session information. A more reliable and secure method for device administration is to use the Secure Shell (SSH) protocol. SSH, which provides secure encryption and strong authentication, is available in two versions:

- **SSH Version 1 (SSHv1)** - This is an improvement over using plaintext Telnet, but some fundamental flaws exist in its implementation, so it should be avoided in favor of SSHv2.
- **SSH Version 2 (SSHv2)** - This is a complete rework and stronger version of SSH that is not compatible with SSHv1. SSHv2 has many benefits and closes a security hole that is found in SSH version 1. SSH version 2 is certified under the National Institute of Standards and Technology (NIST) Federal Information Processing Standards (FIPS) 140-1 and 140-2 U.S. cryptographic standards and should be used where feasible.
 - *To force the IOS SSH server to disable SSHv1 and accept only SSHv2 connections, enter the command `ip ssh version 2` under global configuration mode.*

Configuring vty Access Using SSH

The steps needed to configure SSH on an IOS device are as follows:

Step 1. Configure a **hostname** other than Router by using the command `hostname {hostname name}`.

Step 2. Configure a **domain name** by using the command `ip domain-name {domain-name}`.

Step 3. Generate **crypto keys** by using the command `crypto key generate rsa`. When entering this command, you are prompted to enter a modulus length. The longer the modulus, the stronger the security. However, a longer modulus takes longer to generate. The modulus length needs to be **at least 768** bits for SSHv2.

Example 26-16 Configuring vty Access Using SSH

```
R1(config)# hostname R1
R1(config)# username cisco secret cisco
R1(config)# ip domain-name cisco.com
R1(config)# crypto key generate rsa
The name for the keys will be: R1.cisco.com
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 768
% Generating 768 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

R1(config)#
*May  8 20:44:48.319: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# end
```

Password Combinations and Access

No.	Console password	Vty password	Local access	Remote access
1	No	No	Yes	No
2	No	Yes	Yes	Yes No transit into privileged mode *
3	Yes	No	Yes (password)	No
4	Yes	Yes	Yes (password)	Yes Privileged mode can be set by console password *

*) If `enable password` not set

Aux Port and Exec Timeout

Some devices have an auxiliary (aux) port available for remote administration through a dialup modem connection. In most cases, the aux port should be disabled by using the command `no exec` under line `aux 0`.

By default, an idle EXEC session is not terminated, which poses an enormous security risk. The command `exec-timeout {minutes}{seconds}` under line configuration mode can be used to disconnect idle user sessions. The default setting is 10 minutes.

Example 26-17 shows a configuration in which the `exec-timeout` for the console line is configured to time out after 5 minutes of inactivity and 2 minutes and 30 seconds for the vty lines.

Example 26-17 *Configuring EXEC Timeout*

```
line con 0
  exec-timeout 5 0
line vty 0 4
  exec-timeout 2 30
```

```
exec-timeout 0 – no timeout
```

Absolute Timeout

The command **absolute-timeout** *{minutes}* under line configuration mode **terminates an EXEC session after the specified timeout period has expired, even if the connection is being used at the time of termination.**

It is recommended to use it in combination with the command **logout-warning** *{seconds}* under line configuration mode to display a “line termination” warning to users about an impending forced timeout.

Example 26-18 shows the commands **absolute-timeout** and **logout-warning** configured on the vty lines.

Example 26-18 *Configuring Absolute Timeout*

```
line vty 4
  exec-timeout 2 0
  absolute-timeout 10
  logout-warning 20
```

Authentication, Authorization, and Accounting (AAA)

AAA is an architectural framework for enabling a set of three independent security functions:

- Authentication
- Authorization
- Accounting

AAA is commonly used in the networking industry for the following two use cases:

- Network device access control
- Secure network access control

AAA Framework

AAA is an architectural framework for enabling a set of three independent security functions:

- **Authentication** - Enables a user to be identified and verified prior to being granted access to a network device and/or network services.
- **Authorization** - Defines the access privileges and restrictions to be enforced for an authenticated user.
- **Accounting** - Provides the ability to track and log user access, including user identities, start and stop times, executed commands (that is, CLI commands), and so on. In other words, it maintains a security log of events.

There are many AAA protocols available, but the two most popular ones are **Remote Authentication Dial-In User Service (RADIUS)** and **Terminal Access Controller Access-Control System Plus (TACACS+)**.

AAA Use Cases

AAA is commonly used in the networking industry for the following two use cases:

- **Network device access control** - As described earlier in this chapter, Cisco IOS provides local features for simple device access control, such as local username-based authentication and line password authentication. However, these features do not provide the same degree of access control and scalability that is possible with AAA. For this reason, AAA is the recommended method for access control. **TACACS+** is the protocol of choice for network device access control.
- **Secure network access control** - AAA can be used to obtain the identity of a device or user before that device or user is allowed to access to the network. **RADIUS** is the preferred protocol for secure network access.

TACACS+

Cisco developed TACACS+ and released it as an open standard in the early 1990s. Although TACACS+ is mainly used for AAA device access control, it is possible to use it for some types of AAA network access.

TACACS → XTACACS → TACACS+

The TACACS+ protocol uses Transmission Control Protocol (TCP) port 49 for communication between the TACACS+ clients and the TACACS+ server.

Figure 26-1 shows an end user who can access a Cisco switch using Telnet, SSH, or the console. The Cisco switch is acting as a TACACS+ client that communicates with the TACACS+ server using the TACACS+ protocol.

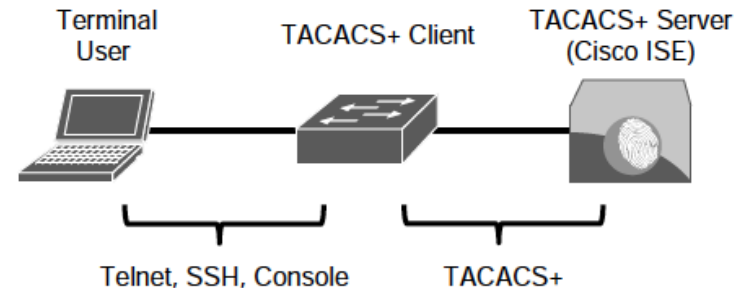


Figure 26-1 TACACS+ Client/Server Communication

RADIUS

One of the **key differentiators of TACACS+** is its capability to **separate authentication, authorization, and accounting** into independent functions. This is why TACACS+ is so commonly used for device administration instead of RADIUS, even though RADIUS is capable of providing network device access control.

RADIUS is an IETF standard AAA protocol. RADIUS is the AAA protocol of choice for secure network access. The reason for this is that RADIUS is the AAA transport protocol for Extensible Authentication Protocol (EAP), while TACACS+ does not support this functionality.

Another major difference between TACACS+ and RADIUS is that RADIUS needs to return all authorization parameters in a single reply, while TACACS+ can request authorization parameters separately and multiple times throughout a session.

RADIUS & TACACS+ Comparison

Table 26-4 provides a summary comparison of RADIUS and TACACS+.

Component	RADIUS	TACACS+
Protocol and port(s) used	Cisco's implementation: <ul style="list-style-type: none"> • UDP: port 1645 (authentication and authorization) • UDP: port 1646 (accounting) Industry standard: • UDP: port 1812 (authentication and authorization) • UDP: port 1813 (accounting) 	TCP: port 49
Encryption	<ul style="list-style-type: none"> • Encrypts only the password field • Supports EAP for 802.1x authentication 	<ul style="list-style-type: none"> • Encrypts the entire payload • Does not support EAP
Authentication and authorization	<ul style="list-style-type: none"> • Combines authentication and authorization • Cannot be used to authorize which CLI commands can be executed individually 	<ul style="list-style-type: none"> • Separates authentication and authorization • Can be used for CLI command authorization
Accounting	Does not support network device CLI command accounting	Supports network device CLI command accounting
Primary Use	Secure network access	Network device access control

Configuring AAA for Network Device Access Control

There are two parts to configuring TACACS+:

- The configuration of the device itself
- The configuration of the TACACS+ AAA server (for example, Cisco ISE)

The following steps are for configuring an IOS device with TACACS+ for device access control. Configuration for the TACACS+ server is not included here because it is beyond the scope of this book:

Step 1. Create a local user with full privilege for fallback or to avoid being locked out after enabling AAA by using the command

```
username {username} privilege 15 algorithm-type {md5 | sha256 | scrypt}  
secret {password}
```

Configuring AAA for Network Device Access Control (Cont.)

Step 2. Enable AAA functions on by using with the command `aaa new-model`.

Step 3. Add a TACACS+ server using one of these methods, depending on the IOS version:

- To add a TACACS+ server on IOS versions prior to 15.x, use the command
- To add a TACACS+ server on IOS versions 15.x and later, use the following commands:

```
tacacs-server host { hostname | host-ip-address } key key-string
```

```
tacacs server name
```

```
address ipv4 { hostname | host-ip-address }
```

```
key key-string
```

Step 4. Create an AAA group by using the following commands:

```
aaa group server tacacs+ group-name
```

```
server name server-name.
```

This creates an AAA group that includes the TACACS+ servers that are added to the group with the `server name` command.

Configuring AAA for Network Device Access Control (Cont.)

Step 5. Enable AAA login authentication by using the command `aaa authentication login { default | custom-list-name } method1 [method2 . . .]` Method lists enable login authentication. The default keyword applies the method lists that follow (method1 [method2 . . .]) to all lines (cty, tty, aux, and so on). The *custom list-name* CLI assigns a custom name for the method lists that follow it. To apply a custom list to a line, use the command `login authentication custom-list-name` under the line configuration mode. Method lists are applied sequentially from left to right.

Step 6. Enable AAA authorization for EXEC by using the command `aaa authorization exec { default | custom-list-name } method1 [method2 . . .]` This command enables EXEC shell authorization for all lines except the console line.

Step 7. Enable AAA authorization for the console by using the command `aaa authorization console`. Authorization for the console is disabled by default to prevent inexperienced users from locking themselves out.

Configuring AAA for Network Device Access Control (Cont.)

Step 8. Enable AAA command authorization by using the command **aaa authorization commands** *{privilege level}* { **default** | *custom-list-name* } *method1* [*method2* . . .]

Step 9. Enable command authorization in global configuration mode (and all global configuration submodes) by using the command **aaa authorization config-commands**

Step 10. Enable login accounting by using the command **aaa accounting exec** { **default** | *custom-list-name* } *method1* [*method2* . . .]

Step 11. Enable command accounting by using the command **aaa accounting commands** *{privilege level}* { **default** | *custom-list-name* } *method1* [*method2* . . .]

AAA Configuration For Device Control Example

Example 26-19 shows a common AAA IOS configuration for device access control.

Example 26-19 Common AAA Configuration for Device Access Control

```
aaa new-model

tacacs server ISE-PRIMARY
  address 10.10.10.1
  key my.S3cR3t.k3y

tacacs server ISE-SECONDARY
  address 20.20.20.1
  key my.S3cR3t.k3y

aaa group server tacacs+ ISE-TACACS+
  server name ise-primary
  server name ise-secondary

aaa authentication login default group ISE-TACACS+ local
aaa authentication login CONSOLE-CUSTOM-AUTHENTICATION-LIST local line enable
aaa authentication enable default group ISE-TACACS+ enable
aaa authorization exec default group ISE-TACACS+ if-authenticated
aaa authorization exec CONSOLE-CUSTOM-EXEC-AUTHORIZATION-LIST none
aaa authorization commands 0 CONSOLE-CUSTOM-COMMAND-AUTHORIZATION-LIST none
aaa authorization commands 1 CONSOLE-CUSTOM-COMMAND-AUTHORIZATION-LIST none
aaa authorization commands 15 CONSOLE-CUSTOM-COMMAND-AUTHORIZATION-LIST none
aaa authorization commands 0 default group ISE-TACACS+ if-authenticated
aaa authorization commands 1 default group ISE-TACACS+ if-authenticated
aaa authorization commands 15 default group ISE-TACACS+ if-authenticated
aaa authorization console
aaa authorization config-commands
aaa accounting exec default start-stop group ISE-TACACS+
aaa accounting commands 0 default start-stop group ISE-TACACS+
aaa accounting commands 1 default start-stop group ISE-TACACS+
aaa accounting commands 15 default start-stop group ISE-TACACS+
```

```
line con 0
  authorization commands 0 CONSOLE-CUSTOM-COMMAND-AUTHORIZATION-LIST
  authorization commands 1 CONSOLE-CUSTOM-COMMAND-AUTHORIZATION-LIST
  authorization commands 15 CONSOLE-CUSTOM-COMMAND-AUTHORIZATION-LIST
  authorization exec CONSOLE-CUSTOM-EXEC-AUTHORIZATION-LIST
  privilege level 15
  login authentication CONSOLE-CUSTOM-AUTHENTICATION-LIST

line vty 0 4
  <uses default method-lists for AAA>
```

Apart from the IOS configuration, the AAA server also needs to be configured with the AAA client information (hostname, IP address, and key), the login credentials for the users, and the commands the users are authorized to execute on the device.

Authentication, Authorization, and Accounting (AAA)

Verifying AAA Configuration

Example 26-20 demonstrates SSH sessions being initiated from R2 into R1, using the netadmin and netops accounts.

The netadmin account was configured in the AAA server with privilege 15, and netops was configured with privilege 1. The netadmin account has access to the full set of commands, while netops is very limited.

Example 26-20 Verifying AAA Configuration

```
! Establish SSH session from R2 into R1 using netadmin account
R2# ssh netadmin@10.12.1.1
Password:
R1# show privilege
Current privilege level is 15
R1#
R1# configure terminal
R1(config)#

! Establish SSH session from R2 into R1 using netops account

R2# ssh netops@192.168.1.26
Password:
R1> show privilege
Current privilege level is 1
R1> show version
Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(3)M2,
RELEASE SOFTWARE (fc2)
! Output Omitted for Brevity

R1> show running-config
Command authorization failed.

R1> enable
Command authorization failed.
```

Zone-Based Firewall (ZBFW)

- Cisco Zone-Based Firewall (ZBFW) is the latest integrated stateful firewall technology included in IOS.
- ZBFW uses a flexible and straightforward approach to providing security by establishing security zones.
- A zone establishes a security border on the network and defines acceptable traffic that is allowed to pass between zones.
- By default, interfaces in the same security zone can communicate freely with each other, but interfaces in different zones cannot communicate with each other.

The Self & Default Zones

The **self zone** is a **system-level zone** and includes **all the routers' IP addresses**. By default, traffic to and from this zone is permitted to support management (for example, SSH protocol, SNMP) and control plane (for example, EIGRP, BGP) functions. After a policy is applied to the self zone and other security zone, interzone communication must be explicitly defined.

The **default zone** is a **system-level zone**, and **any interface** that is **not a member of another security zone** is placed in this zone automatically. Upon initialization of this zone, any interface not associated to a security zone is placed in this zone. When the unassigned interfaces are in the default zone, a policy map can be created between the two security zones.

ZBFW Configuration

ZBFW is configured in five steps:

Step 1. Configure the security zones by using the command `zone security zone-name`. A zone needs to be created for the outside zone (the Internet). The self zone is defined automatically.

Example 26-21 demonstrates the configuration of a security zone.

Example 26-21 *Defining the Outside Security Zone*

```
Zone security OUTSIDE
description OUTSIDE Zone used for Internet Interface
```

ZBFW Configuration (Cont.)

Step 2. Define the inspection class map. The class map for inspection defines a method for classification of traffic. The class map is configured using the command **classmap type inspect [match-all | match-any] class-name**.

Example 26-22 shows a sample configuration of inspection class maps and their associated ACLs.

Example 26-22 *Inspecting the Class Map Configuration*

```
ip access-list extended ACL-IPSEC
 permit udp any any eq non500-isaicmp
 permit udp any any eq isacmp
ip access-list extended ACL-PING-AND-TRACEROUTE
 permit icmp any any echo
 permit icmp any any echo-reply
 permit icmp any any ttl-exceeded
 permit icmp any any port-unreachable
 permit udp any any range 33434 33463 ttl eq 1
ip access-list extended ACL-ESP
 permit esp any any
ip access-list extended ACL-DHCP-IN
 permit udp any eq bootps any eq bootpc
ip access-list extended ACL-GRE
 permit gre any any
!
class-map type inspect match-any CLASS-OUTSIDE-TO-SELF-INSPECT
 match access-group name ACL-IPSEC
 match access-group name ACL-PING-AND-TRACEROUTE
class-map type inspect match-any CLASS-OUTSIDE-TO-SELF-PASS
 match access-group name ACL-ESP
 match access-group name ACL-DHCP-IN
 match access-group name ACL-GRE
```

ZBFW Configuration (Cont.)

The configuration of inspect class maps can be verified with the command **show class-map type inspect** [*class-name*], as shown in Example 26-23.

Example 26-23 *Verifying the Inspect Class Map Configuration*

```
R1# show class-map type inspect
Class Map type inspect match-any CLASS-OUTSIDE-TO-SELF-PASS (id 2)
  Match access-group name ACL-ESP
  Match access-group name ACL-DHCP-IN
  Match access-group name ACL-GRE

Class Map type inspect match-any CLASS-OUTSIDE-TO-SELF-INSPECT (id 1)
  Match access-group name ACL-IPSEC
  Match access-group name ACL-PING-AND-TRACEROUTE
```

ZBFW Configuration (Cont.)

Step 3. Define the inspection policy map, which applies firewall policy actions to the class maps defined in the policy map.

The policy map is then associated to a zone pair.

The inspection policy map is defined with the command `policy-map type inspect policy-name`.

After the policy map is defined, the various class maps are defined with the command `class type inspect class-name`.

ZBFW Configuration (Cont.)

Under the class map, the firewall action is defined with these commands:

- **drop** [log]: This default action silently discards packets that match the class map. The log keyword adds syslog information that includes source and destination information (IP address, port, and protocol).
- **pass** [log]: This action makes the router forward packets from the source zone to the destination zone. Packets are forwarded in only one direction. A policy must be applied for traffic to be forwarded in the opposite direction. The pass action is useful for protocols like IPsec, Encapsulating Security Payload (ESP), and other inherently secure protocols with predictable behavior. The optional log keyword adds syslog information that includes the source and destination information.
- **inspect**: The inspect action offers state-based traffic control. The router maintains connection/session information and permits return traffic from the destination zone without

ZBFW Configuration (Cont.)

Example 26-24 demonstrates the configuration of the inspect policy map. Notice that in the class default class, the **drop** command does not include the log keyword because of the potential to fill up the syslog.

The inspection policy map can be verified with the command **show policy-map type inspect** *[policy-name]*, as shown in Example 26-25.

Example 26-24 *Configuring the Inspection Policy Map*

```
policy-map type inspect POLICY-OUTSIDE-TO-SELF
  class type inspect CLASS-OUTSIDE-TO-SELF-INSPECT
    inspect
  class type inspect CLASS-OUTSIDE-TO-SELF-PASS
    pass
  class class-default
    drop
```

Example 26-25 *Verifying the Inspection Policy Map*

```
R1# show policy-map type inspect
Policy Map type inspect POLICY-OUTSIDE-TO-SELF
  Class CLASS-OUTSIDE-TO-SELF-INSPECT
    Inspect
  Class CLASS-OUTSIDE-TO-SELF-PASS
    Pass
  Class class-default
    Drop
```

ZBFW Configuration (Cont.)

Step 4. Apply a policy map to a traffic flow source to a destination by using the command `zone-pair security zone-pair-name source source-zone-name destination destination-zone-name`. The inspection policy map is then applied to the zone pair with the command `service-policy type inspect policy-name`. Traffic is statefully inspected between the source and destination, and return traffic is allowed.

Example 26-26 defines the zone pairs and associates the policy map to the zone pair.

Example 26-26 *Configuring the ZBFW Zone Pair*

```
zone-pair security OUTSIDE-TO-SELF source OUTSIDE destination self
  service-policy type inspect POLICY-OUTSIDE-TO-SELF
```

ZBFW Configuration (Cont.)

Step 5. Apply the security zones to the appropriate interfaces. An interface is assigned to the appropriate zone by entering the interface configuration submode with the command `interface interface-id` and associating the interface to the correct zone with the command `zone-member security zone-name`, as defined in step 1.

Example 26-27 demonstrates the outside security zone being associated to the Internet-facing interface GigabitEthernet 0/2.

Example 26-27 *Applying the Security Zone to the Interface*

```
interface GigabitEthernet 0/2
  zone-member security OUTSIDE
```

Verifying the Outside-to-Self Policy

Now that the outside-to-self policy has been fully defined, traffic statistics can be viewed with the command **show policy-map type inspect zone-pair [zone-pair-name]**.

Example 26-28 demonstrates the verification of the configured ZBFW policy.

Example 26-28 *Verifying the Outside-to-Self Policy*

```
R1# show policy-map type inspect zone-pair

policy exists on zp OUTSIDE-TO-SELF
Zone-pair: OUTSIDE-TO-SELF

Service-policy inspect : POLICY-OUTSIDE-TO-SELF

Class-map: CLASS-OUTSIDE-TO-SELF-INSPECT (match-any)
  Match: access-group name ACL-IPSEC
    2 packets, 208 bytes
    30 second rate 0 bps
  Match: access-group name ACL-PING-AND-TRACEROUTE
    0 packets, 0 bytes
    30 second rate 0 bps
```

```
Inspect
  Packet inspection statistics [process switch:fast switch]
  udp packets: [4:8]

  Session creations since subsystem startup or last reset 2
  Current session counts (estab/half-open/terminating) [0:0:0]
  Maxever session counts (estab/half-open/terminating) [2:1:0]
  Last session created 00:03:39
  Last statistic reset never
  Last session creation rate 0
  Maxever session creation rate 2
  Last half-open session total 0
  TCP reassembly statistics
  received 0 packets out-of-order; dropped 0
  peak memory usage 0 KB; current usage: 0 KB
  peak queue length 0
```

```
Class-map: CLASS-OUTSIDE-TO-SELF-PASS (match-any)
  Match: access-group name ACL-ESP
    186 packets, 22552 bytes
    30 second rate 0 bps
  Match: access-group name ACL-DHCP-IN
    1 packets, 308 bytes
    30 second rate 0 bps
  Match: access-group name ACL-GRE
    0 packets, 0 bytes
    30 second rate 0 bps
  Pass
    187 packets, 22860 bytes

Class-map: class-default (match-any)
  Match: any
  Drop
    30 packets, 720 bytes
```

ACL Counters from the Inspect Class Maps

Even though the ACLs are not used for blocking traffic, the counters do increase as packets match the ACL entries for the inspect class maps, as demonstrated in Example 26-29.

Example 26-29 *ACL Counters from the Inspect Class Maps*

```
R1# show ip access
Extended IP access list ACL-DHCP-IN
  10 permit udp any eq bootps any eq bootpc (1 match)
Extended IP access list ACL-ESP
  10 permit esp any any (170 matches)
Extended IP access list ACL-GRE
  10 permit gre any any
Extended IP access list ACL-IPSEC
  10 permit udp any any eq non500-isakmp
  20 permit udp any any eq isakmp (2 matches)
Extended IP access list ACL-PING-AND-TRACEROUTE
  10 permit icmp any any echo
  20 permit icmp any any echo-reply
  30 permit icmp any any ttl-exceeded
  40 permit icmp any any port-unreachable
  50 permit udp any any range 33434 33463 ttl eq 1
```

Zone-Based Firewall (ZBFW)

Verifying ZBFW

After the outside-to-self policy has been defined, it is time to verify connectivity to the internet, as shown in Example 26-30. Notice here that a simple ping from R1 to one of Google's Public DNS IP addresses 8.8.8.8 is failing.

Example 26-30 *Verifying Outside Connectivity*

```
R1# ping 8.8.8.8
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Configuring the Self-to-Outside Policy

The reason for the packet failure is that the router needs to allow locally originated packets with a self-to-outside policy.

Example 26-31 demonstrates the configuration for the self-to-outside policy. ACL-IPSEC and ACL-ESP are reused from the outside-to-self policy.

Now that the second policy has been configured, R1 can successfully ping 8.8.8.8, as shown in Example 26-32.

Example 26-32 *Successful ping Test Between R1 and Google's Public DNS 8.8.8.8*

```
R31-Spoke# ping 8.8.8.8
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Example 26-31 *Configuring the Self-to-Outside Policy*

```
ip access-list extended ACL-DHCP-OUT
permit udp any eq bootpc any eq bootps
!
ip access-list extended ACL-ICMP
permit icmp any any
!
class-map type inspect match-any CLASS-SELF-TO-OUTSIDE-INSPECT
match access-group name ACL-IPSEC
match access-group name ACL-ICMP

class-map type inspect match-any CLASS-SELF-TO-OUTSIDE-PASS
match access-group name ACL-ESP
match access-group name ACL-DHCP-OUT
!
policy-map type inspect POLICY-SELF-TO-OUTSIDE
class type inspect CLASS-SELF-TO-OUTSIDE-INSPECT
inspect
class type inspect CLASS-SELF-TO-OUTSIDE-PASS
pass
class class-default
drop log
!
zone-pair security SELF-TO-OUTSIDE source self destination OUTSIDE
service-policy type inspect POLICY-SELF-TO-OUTSIDE
```

Control Plane Policing (CoPP)

- A control plane policing (CoPP) policy is a QoS policy that is applied to traffic to or sourced by the router's control plane CPU.
- CoPP policies are used to limit known traffic to a given rate while protecting the CPU from unexpected extreme rates of traffic that could impact the stability of the router.
- Typical CoPP implementations use only an input policy that allows traffic to the control plane to be policed to a desired rate.
- The CoPP policy is then implemented to limit traffic to the control plane CPU to a specific rate for each class.
- The QoS police command uses **conform**, **exceed**, and **violate** actions, which can be configured to transmit or drop traffic.

Control Plane Policing (CoPP)

Configuring ACLs for CoPP

After the network traffic has been identified, ACLs can be built for matching in a class map.

Example 26-33 demonstrates a list of ACLs matching traffic **identified** by network documentation etc.

Example 26-33 *Configuring an Access List for CoPP*

```
ip access-list extended ACL-CoPP-ICMP
 permit icmp any any echo-reply
 permit icmp any any ttl-exceeded
 permit icmp any any unreachable
 permit icmp any any echo
 permit udp any any range 33434 33463 ttl eq 1
!
```

```
ip access-list extended ACL-CoPP-IPsec
 permit esp any any
 permit gre any any
 permit udp any eq isakmp any eq isakmp
 permit udp any any eq non500-isakmp
 permit udp any eq non500-isakmp any
!
ip access-list extended ACL-CoPP-Initialize
 permit udp any eq bootps any eq bootpc
!
ip access-list extended ACL-CoPP-Management
 permit udp any eq ntp any
 permit udp any any eq snmp
 permit tcp any any eq 22
 permit tcp any eq 22 any established
!
ip access-list extended ACL-CoPP-Routing
 permit tcp any eq bgp any established
 permit eigrp any host 224.0.0.10
 permit ospf any host 224.0.0.5
 permit ospf any host 224.0.0.6
 permit pim any host 224.0.0.13
 permit igmp any any
```

Class Configuration for CoPP

The class configuration for CoPP uses the ACLs to match known protocols being used and is demonstrated in Example 26-34.

Example 26-34 *Class Configuration for CoPP*

```
class-map match-all CLASS-CoPP-IPsec
  match access-group name ACL-CoPP-IPsec
class-map match-all CLASS-CoPP-Routing
  match access-group name ACL-CoPP-Routing
class-map match-all CLASS-CoPP-Initialize
  match access-group name ACL-CoPP-Initialize
class-map match-all CLASS-CoPP-Management
  match access-group name ACL-CoPP-Management
class-map match-all CLASS-CoPP-ICMP
  match access-group name ACL-CoPP-ICMP
```

Policy Configuration for CoPP

The policy map for how the classes operate shows how to police traffic to a given rate in order to minimize any ability to overload the router.

In order to guarantee that CoPP does not introduce issues, the **violate** action is set to **transmit** for all the vital classes until a baseline for normal traffic flows is established.

Example 26-35 shows the CoPP policy.

Example 26-35 *Policy Configuration for CoPP*

```
policy-map POLICY-CoPP
  class CLASS-CoPP-ICMP
    police 8000 conform-action transmit exceed-action transmit
      violate-action drop
  class CLASS-CoPP-IPsec
    police 64000 conform-action transmit exceed-action transmit
      violate-action transmit
  class CLASS-CoPP-Initialize
    police 8000 conform-action transmit exceed-action transmit
      violate-action drop
  class CLASS-CoPP-Management
    police 32000 conform-action transmit exceed-action transmit
      violate-action transmit
  class CLASS-CoPP-Routing
    police 64000 conform-action transmit exceed-action transmit
      violate-action transmit
  class class-default
    police 8000 conform-action transmit exceed-action transmit
      violate-action drop
```

Control Plane Policing (CoPP)

Applying the Policy for CoPP

The CoPP policy map needs to be applied to the control plane with the command **service-policy {input|output} *policy-name*** under control plane configuration mode, as demonstrated in Example 26-36.

Example 26-36 *Applying the Policy for CoPP*

```
control-plane
service-policy input POLICY-CoPP
```

Verifying the Policy for CoPP

After the policy map has been applied to the control plane, it needs to be verified. In Example 26-37, traffic matching CLASS-CoPP-Routing has exceeded the configured rate.

Example 26-37 Verifying the Policy for CoPP

```
R1# show policy-map control-plane input
Control Plane

Service-policy input: POLICY-CoPP

Class-map: CLASS-CoPP-ICMP (match-all)
 154 packets, 8912 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
 Match: access-group name ACL-CoPP-ICMP
 police:
   cir 8000 bps, bc 1500 bytes, be 1500 bytes
   conformed 154 packets, 8912 bytes; actions:
     transmit
   exceeded 0 packets, 0 bytes; actions:
     transmit
   violated 0 packets, 0 bytes; actions:
     drop
   conformed 0000 bps, exceeded 0000 bps, violated 0000 bps
```

```
Class-map: CLASS-CoPP-IPsec (match-all)
 0 packets, 0 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
 Match: access-group name ACL-CoPP-IPsec
 police:
   cir 64000 bps, bc 2000 bytes, be 2000 bytes
   conformed 0 packets, 0 bytes; actions:
     transmit
   exceeded 0 packets, 0 bytes; actions:
     transmit
   violated 0 packets, 0 bytes; actions:
     transmit
   conformed 0000 bps, exceeded 0000 bps, violated 0000 bps

Class-map: CLASS-CoPP-Initialize (match-all)
 0 packets, 0 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
```

Verifying the Policy for CoPP (Cont.)

```
Match: access-group name ACL-CoPP-Initialize
police:
  cir 8000 bps, bc 1500 bytes, be 1500 bytes
  conformed 0 packets, 0 bytes; actions:
    transmit
  exceeded 0 packets, 0 bytes; actions:
    transmit
  violated 0 packets, 0 bytes; actions:
    drop
  conformed 0000 bps, exceeded 0000 bps, violated 0000 bps
```

```
Class-map: CLASS-CoPP-Management (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
Match: access-group name ACL-CoPP-Management
police:
  cir 32000 bps, bc 1500 bytes, be 1500 bytes
  conformed 0 packets, 0 bytes; actions:
    transmit
  exceeded 0 packets, 0 bytes; actions:
    transmit
  violated 0 packets, 0 bytes; actions:
    transmit
  conformed 0000 bps, exceeded 0000 bps, violated 0000 bps
```

```
Class-map: CLASS-CoPP-Routing (match-all)
  92 packets, 123557 bytes
  5 minute offered rate 4000 bps, drop rate 0000 bps
Match: access-group name ACL-CoPP-Routing
police:
  cir 64000 bps, bc 2000 bytes, be 2000 bytes
  conformed 5 packets, 3236 bytes; actions:
    transmit
  exceeded 1 packets, 1383 bytes; actions:
    transmit
  violated 86 packets, 118938 bytes; actions:
    transmit
  conformed 1000 bps, exceeded 1000 bps, violated 4000 bps
```

```
Class-map: class-default (match-any)
  56 packets, 20464 bytes
  5 minute offered rate 1000 bps, drop rate 0000 bps
Match: any
police:
  cir 8000 bps, bc 1500 bytes, be 1500 bytes
  conformed 5 packets, 2061 bytes; actions:
    transmit
  exceeded 0 packets, 0 bytes; actions:
    transmit
  violated 0 packets, 0 bytes; actions:
    drop
  conformed 0000 bps, exceeded 0000 bps, violated 0000 bps
```

Device Hardening

In addition to providing device access control and protection, disabling unused services and features, hardening a router reduces the amount of router CPU and memory utilization that would be required to process those packets.

Device Hardening

The following is a list of additional commands that can be used to harden a router. All interface-specific commands are applied only to the interface connected to the public network.

- **Disable topology discovery tools** - Tools such as Cisco Discovery Protocol (CDP) and Link Layer Discovery Protocol (LLDP) can provide unnecessary information to routers outside your control. The services can be disabled with the interface parameter commands **no cdp enable**, **no lldp transmit**, and **no lldp receive**.
- **Disable TCP and UDP small services** - The commands **service tcp-keepalive-in** and **service tcp-keepalive-out** ensure that devices send TCP keepalives for inbound/outbound TCP sessions. This ensures that the device on the remote end of the connection is still accessible and that half-open or orphaned connections are removed from the local device.
- **Disable IP redirect services** - An ICMP redirect is used to inform a device of a better path to the destination network. An IOS device sends an ICMP redirect if it detects network traffic hairpinning on it. This behavior is disabled with the interface parameter command **no ip redirects**.
- **Disable proxy Address Resolution Protocol (ARP)** - Proxy ARP is a technique that a router uses to answer ARP requests intended for a different router. The router fakes its identity and sends out an ARP response for the router that is responsible for that network.

Device Hardening (Cont.)

- **Disable service configuration** - Cisco devices support automatic configuration from remote devices through TFTP and other methods. This service should be disabled with the command **no service config**.
- **Disable the Maintenance Operation Protocol (MOP) service** - The MOP service is not needed and should be disabled globally with the command **no mop enabled** and with the interface parameter command **no mop enabled**.
- **Disable the packet assembler/disassembler (PAD) service** - The PAD service is used for X.25 and is not needed. It can be disabled with the **command no service pad**.

