Cisco Guide to Harden Cisco IOS Devices

Document ID: 13608

Contributed by Shashank Singh, Cisco TAC Engineer.

Jan 06, 2016

Contents

Introduction
Prerequisites
    Requirements
    Components Used
Secure Operations
    Monitor Cisco Security Advisories and Responses
    Leverage Authentication, Authorization, and Accounting
    Centralize Log Collection and Monitoring
    Use Secure Protocols When Possible
    Gain Traffic Visibility with NetFlow
    Configuration Management
Management Plane
    General Management Plane Hardening
        Password Management
        Enhanced Password Security
        Login Password Retry Lockout
        No Service Password-Recovery
        Disable Unused Services
        EXEC Timeout
        Keepalives for TCP Sessions
        Management Interface Use
        Memory Threshold Notifications
        CPU Thresholding Notification
        Reserve Memory for Console Access
        Memory Leak Detector
        Buffer Overflow: Detection and Correction of Redzone Corruption
        Enhanced Crashinfo File Collection
        Network Time Protocol
    Limit Access to the Network with Infrastructure ACLs
        ICMP Packet Filtering
        Filter IP Fragments
        ACL Support for Filtering IP Options
        ACL Support to Filter on TTL Value
    Secure Interactive Management Sessions
        Management Plane Protection
        Control Plane Protection
        Encrypt Management Sessions
        SSHv2
        SSHv2 Enhancements for RSA Keys
        Console and AUX Ports
        Control vty and tty Lines
        Control Transport for vty and tty Lines
        Warning Banners
    Authentication, Authorization, and Accounting

    - TACACS+ Authentication
    - Authentication Fallback
    - Use of Type 7 Passwords
    - TACACS+ Command Authorization
    - TACACS+ Command Accounting
    - Redundant AAA Servers
  - Fortify the Simple Network Management Protocol
    - SNMP Community Strings
    - SNMP Community Strings with ACLs
    - Infrastructure ACLs
    - SNMP Views
    - SNMP Version 3
    - Management Plane Protection
  - Logging Best Practices
    - Send Logs to a Central Location
    - Logging Level
    - Do Not Log to Console or Monitor Sessions
    - Use Buffered Logging
    - Configure Logging Source Interface
    - Configure Logging Timestamps
  - Cisco IOS Software Configuration Management
    - Configuration Replace and Configuration Rollback
    - Exclusive Configuration Change Access
    - Cisco IOS Software Resilient Configuration
    - Digitally Signed Cisco Software
    - Configuration Change Notification and Logging
- **Control Plane**
  - General Control Plane Hardening
    - IP ICMP Redirects
    - ICMP Unreachables
    - Proxy ARP
  - Limit CPU Impact of Control Plane Traffic
    - Understand Control Plane Traffic
    - Infrastructure ACLs
    - Receive ACLs
    - Control Plane Policing
    - Control Plane Protection
    - Hardware Rate Limiters
  - Secure BGP
    - TTL-based Security Protections
    - BGP Peer Authentication with MD5
    - Configure Maximum Prefixes
    - Filter BGP Prefixes with Prefix Lists
    - Filter BGP Prefixes with Autonomous System Path Access Lists
  - Secure Interior Gateway Protocols
    - Routing Protocol Authentication and Verification with Message Digest 5
    - Passive-Interface Commands
    - Route Filtering
    - Routing Process Resource Consumption
  - Secure First Hop Redundancy Protocols
- **Data Plane**
  - General Data Plane Hardening
    - IP Options Selective Drop
    - Disable IP Source Routing

        - Disable ICMP Redirects
        - Disable or Limit IP Directed Broadcasts
    - Filter Transit Traffic with Transit ACLs
        - ICMP Packet Filtering
        - Filter IP Fragments
        - ACL Support for Filtering IP Options
    - Anti-Spoofing Protections
        - Unicast RPF
        - IP Source Guard
        - Port Security
        - Dynamic ARP Inspection
        - Anti-Spoofing ACLs
    - Limit CPU Impact of Data Plane Traffic
        - Features and Traffic Types that Impact the CPU
        - Filter on TTL Value
        - Filter on the Presence of IP Options
        - Control Plane Protection
    - Traffic Identification and Traceback
        - NetFlow
        - Classification ACLs
    - Access Control with VLAN Maps and Port Access Control Lists
        - Access Control with VLAN Maps
        - Access Control with PACLs
        - Access Control with MAC
    - Private VLAN Use
        - Isolated VLANs
        - Community VLANs
        - Promiscuous Ports
- **Conclusion**
- **Acknowledgments**
- **Appendix: Cisco IOS Device Hardening Checklist**
    - Management Plane
    - Control Plane
    - Data Plane

Introduction

This document contains information to help you secure your Cisco IOS® system devices, which increases the overall security of your network. Structured around the three planes into which functions of a network device can be categorized, this document provides an overview of each included feature and references to related documentation.

The three functional planes of a network - the management plane, control plane, and data plane - each provide different functionality that needs to be protected.

- **Management Plane** - The management plane manages traffic that is sent to the Cisco IOS device and is made up of applications and protocols such as Secure Shell (SSH) and Simple Network Management Protocol (SNMP).
- **Control Plane** - The control plane of a network device processes the traffic that is paramount to maintain the functionality of the network infrastructure. The control plane consists of applications and protocols between network devices, which includes the Border Gateway Protocol (BGP), as well as the Interior Gateway Protocols (IGPs) such as the Enhanced Interior Gateway Routing Protocol (EIGRP) and Open Shortest Path First (OSPF).

- **Data Plane** - The data plane forwards data through a network device. The data plane does not include traffic that is sent to the local Cisco IOS device.

The coverage of security features in this document often provides enough detail for you to configure the feature. However, in cases where it does not, the feature is explained in such a way that you can evaluate whether additional attention to the feature is required. Where possible and appropriate, this document contains recommendations that, if implemented, help secure a network.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Secure Operations

Secure network operations is a substantial topic. Although most of this document is devoted to the secure configuration of a Cisco IOS device, configurations alone do not completely secure a network. The operational procedures in use on the network contribute as much to security as the configuration of the underlying devices.

These topics contain operational recommendations that you are advised to implement. These topics highlight specific critical areas of network operations and are not comprehensive.

Monitor Cisco Security Advisories and Responses

The Cisco Product Security Incident Response Team (PSIRT) creates and maintains publications, commonly referred to as PSIRT Advisories, for security-related issues in Cisco products. The method used for communication of less severe issues is the Cisco Security Response. Security advisories and responses are available at http://www.cisco.com/go/psirt.

Additional information about these communication vehicles is available in the Cisco Security Vulnerability Policy.

In order to maintain a secure network, you need to be aware of the Cisco security advisories and responses that have been released. You need to have knowledge of a vulnerability before the threat it can pose to a network can be evaluated. Refer to Risk Triage for Security Vulnerability Announcements for assistance this evaluation process.

Leverage Authentication, Authorization, and Accounting

The Authentication, Authorization, and Accounting (AAA) framework is vital to secure network devices. The AAA framework provides authentication of management sessions and can also limit users to specific, administrator-defined commands and log all commands entered by all users. See the Authentication, Authorization, and Accounting section of this document for more information about how to leverage AAA.

Centralize Log Collection and Monitoring

In order to gain knowledge about existing, emerging, and historic events related to security incidents, your organization must have a unified strategy for event logging and correlation. This strategy must leverage logging from all network devices and use pre-packaged and customizable correlation capabilities.

After centralized logging is implemented, you must develop a structured approach to log analysis and incident tracking. Based on the needs of your organization, this approach can range from a simple diligent review of log data to advanced rule-based analysis.

See the Logging Best Practices section of this document for more information about how to implement logging on Cisco IOS network devices.

Use Secure Protocols When Possible

Many protocols are used in order to carry sensitive network management data. You must use secure protocols whenever possible. A secure protocol choice includes the use of SSH instead of Telnet so that both authentication data and management information are encrypted. In addition, you must use secure file transfer protocols when you copy configuration data. An example is the use of the Secure Copy Protocol (SCP) in place of FTP or TFTP.

See the Secure Interactive Management Sessions section of this document for more information about the secure management of Cisco IOS devices.

Gain Traffic Visibility with NetFlow

NetFlow enables you to monitor traffic flows in the network. Originally intended to export traffic information to network management applications, NetFlow can also be used in order to show flow information on a router. This capability allows you to see what traffic traverses the network in real time. Regardless of whether flow information is exported to a remote collector, you are advised to configure network devices for NetFlow so that it can be used reactively if needed.

More information about this feature is available in the Traffic Identification and Traceback section of this document and at http://www.cisco.com/go/netflow (registered customers only) .

Configuration Management

Configuration management is a process by which configuration changes are proposed, reviewed, approved, and deployed. Within the context of a Cisco IOS device configuration, two additional aspects of configuration management are critical: configuration archival and security.

You can use configuration archives to roll back changes that are made to network devices. In a security context, configuration archives can also be used in order to determine which security changes were made and when these changes occurred. In conjunction with AAA log data, this information can assist in the security auditing of network devices.

The configuration of a Cisco IOS device contains many sensitive details. Usernames, passwords, and the contents of access control lists are examples of this type of information. The repository that you use in order to archive Cisco IOS device configurations needs to be secured. Insecure access to this information can undermine the security of the entire network.

Management Plane

The management plane consists of functions that achieve the management goals of the network. This includes interactive management sessions that use SSH, as well as statistics-gathering with SNMP or NetFlow. When you consider the security of a network device, it is critical that the management plane be protected. If a security incident is able to undermine the functions of the management plane, it can be impossible for you to recover or stabilize the network.

These sections of this document detail the security features and configurations available in Cisco IOS software that help fortify the management plane.

General Management Plane Hardening

The management plane is used in order to access, configure, and manage a device, as well as monitor its operations and the network on which it is deployed. The management plane is the plane that receives and sends traffic for operations of these functions. You must secure both the management plane and control plane of a device, because operations of the control plane directly affect operations of the management plane. This list of protocols is used by the management plane:

- Simple Network Management Protocol
- Telnet
- Secure Shell Protocol
- File Transfer Protocol
- Trivial File Transfer Protocol
- Secure Copy Protocol
- TACACS+
- RADIUS
- NetFlow
- Network Time Protocol
- Syslog

Steps must be taken to ensure the survival of the management and control planes during security incidents. If one of these planes is successfully exploited, all planes can be compromised.

Password Management

Passwords control access to resources or devices. This is accomplished through the definition a password or secret that is used in order to authenticate requests. When a request is received for access to a resource or device, the request is challenged for verification of the password and identity, and access can be granted, denied, or limited based on the result. As a security best practice, passwords must be managed with a TACACS+ or RADIUS authentication server. However, note that a locally configured password for privileged access is still needed in the event of failure of the TACACS+ or RADIUS services. A device can also have other password information present within its configuration, such as an NTP key, SNMP community string, or Routing Protocol key.

The **enable secret** command is used in order to set the password that grants privileged administrative access to the Cisco IOS system. The **enable secret** command must be used, rather than the older **enable password** command. The **enable password** command uses a weak encryption algorithm.

If no enable secret is set and a password is configured for the console tty line, the console password can be used in order to receive privileged access, even from a remote virtual tty (vty) session. This action is almost certainly unwanted and is another reason to ensure configuration of an enable secret.

The **service password-encryption** global configuration command directs the Cisco IOS software to encrypt the passwords, Challenge Handshake Authentication Protocol (CHAP) secrets, and similar data that are saved in its configuration file. Such encryption is useful in order to prevent casual observers from reading passwords, such as when they look at the screen over the muster of an administrator. However, the algorithm used by the **service password-encryption** command is a simple Vigen re cipher. The algorithm is not designed to protect configuration files against serious analysis by even slightly sophisticated attackers and must not be used for this purpose. Any Cisco IOS configuration file that contains encrypted passwords must be treated with the same care that is used for a cleartext list of those same passwords.

While this weak encryption algorithm is not used by the **enable secret** command, it is used by the **enable password** global configuration command, as well as the **password** line configuration command. Passwords of this type must be eliminated and the **enable secret** command or the Enhanced Password Security feature needs to be used.

The **enable secret** command and the Enhanced Password Security feature use Message Digest 5 (MD5) for password hashing. This algorithm has had considerable public review and is not known to be reversible. However, the algorithm is subject to dictionary attacks. In a dictionary attack, an attacker tries every word in a dictionary or other list of candidate passwords in order to find a match. Therefore, configuration files must be securely stored and only shared with trusted individuals.

Enhanced Password Security

The feature Enhanced Password Security, introduced in Cisco IOS Software Release 12.2(8)T, allows an administrator to configure MD5 hashing of passwords for the **username** command. Prior to this feature, there were two types of passwords: Type 0, which is a cleartext password, and Type 7, which uses the algorithm from the Vigen re cipher. The Enhanced Password Security feature cannot be used with protocols that require the cleartext password to be retrievable, such as CHAP.

In order to encrypt a user password with MD5 hashing, issue the **username secret** global configuration command.

```
!

username <name> secret <password>

!
```

Refer to Enhanced Password Security for more information about this feature.

Login Password Retry Lockout

The Login Password Retry Lockout feature, added in Cisco IOS Software Release 12.3(14)T, allows you to lock out a local user account after a configured number of unsuccessful login attempts. Once a user is locked out, their account is locked until you unlock it. An authorized user who is configured with privilege level 15 cannot be locked out with this feature. The number of users with privilege level 15 must be kept to a minimum.

Note that authorized users can lock themselves out of a device if the number of unsuccessful login attempts is reached. Additionally, a malicious user can create a denial of service (DoS) condition with repeated attempts to authenticate with a valid username.

This example shows how to enable the Login Password Retry Lockout feature:

```
!

aaa new-model
aaa local authentication attempts max-fail <max-attempts>
aaa authentication login default local

!

username <name> secret <password>

!
```

This feature also applies to authentication methods such as CHAP and Password Authentication Protocol (PAP).

No Service Password-Recovery

In Cisco IOS Software Release 12.3(14)T and later, the No Service Password-Recovery feature does not allow anyone with console access to insecurely access the device configuration and clear the password. It also does not allow malicious users to change the configuration register value and access NVRAM.

```
!

no service password-recovery

!
```

Cisco IOS software provides a password recovery procedure that relies upon access to ROM Monitor Mode (ROMMON) using the Break key during system startup. In ROMMON, the device software can be reloaded in order to prompt a new system configuration that includes a new password.

The current password recovery procedure enables anyone with console access to access the device and its network. The No Service Password-Recovery feature prevents the completion of the Break key sequence and the entering of ROMMON during system startup.

If **no service password-recovery** is enabled on a device, it is recommended that an offline copy of the device configuration be saved and that a configuration archiving solution be implemented. If it is necessary to recover the password of a Cisco IOS device once this feature is enabled, the entire configuration is deleted.

Refer to Secure ROMMON Configuration Example for more information about this feature.

Disable Unused Services

As a security best practice, any unnecessary service must be disabled. These unneeded services, especially those that use User Datagram Protocol (UDP), are infrequently used for legitimate purposes, but can be used in order to launch DoS and other attacks that are otherwise prevented by packet filtering.

The TCP and UDP small services must be disabled. These services include:

- echo (port number 7)
- discard (port number 9)
- daytime (port number 13)
- chargen (port number 19)

Although abuse of the small services can be avoided or made less dangerous by anti-spoofing access lists, the services must be disabled on any device accessible within the network. The small services are disabled by default in Cisco IOS Software Releases 12.0 and later. In earlier software, the **no service tcp-small-servers** and **no service udp-small-servers** global configuration commands can be issued in order to disable them.

This is a list of additional services that must be disabled if not in use:

- Issue the **no ip finger** global configuration command in order to disable Finger service. Cisco IOS software releases later than 12.1(5) and 12.1(5)T disable this service by default.
- Issue the **no ip bootp server** global configuration command in order to disable Bootstrap Protocol (BOOTP).
- In Cisco IOS Software Release 12.2(8)T and later, issue the **ip dhcp bootp ignore** command in global configuration mode in order to disable BOOTP. This leaves Dynamic Host Configuration Protocol (DHCP) services enabled.
- DHCP services can be disabled if DHCP relay services are not required. Issue the **no service dhcp** command in global configuration mode.
- Issue the **no mop enabled** command in interface configuration mode in order to disable the Maintenance Operation Protocol (MOP) service.
- Issue the **no ip domain-lookup** global configuration command in order to disable Domain Name System (DNS) resolution services.
- Issue the **no service pad** command in global configuration mode in order to disable Packet Assembler/Disassembler (PAD) service, which is used for X.25 networks.
- The HTTP server can be disabled with the **no ip http server** command in global configuration mode, and Secure HTTP (HTTPS) server can be disabled with the **no ip http secure-server** global configuration command.
- Unless Cisco IOS devices retrieve configurations from the network during startup, the **no service config** global configuration command must be used. This prevents the Cisco IOS device from an attempt to locate a configuration file on the network with TFTP.
- Cisco Discovery Protocol (CDP) is a network protocol that is used in order to discover other CDP enabled devices for neighbor adjacency and network topology. CDP can be used by Network Management Systems (NMS) or during troubleshooting. CDP must be disabled on all interfaces that are connected to untrusted networks. This is accomplished with the **no cdp enable** interface command. Alternatively, CDP can be disabled globally with the **no cdp run** global configuration command. Note that CDP can be used by a malicious user for reconnaissance and network mapping.
- Link Layer Discovery Protocol (LLDP) is an IEEE protocol that is defined in 802.1AB. LLDP is similar to CDP. However, this protocol allows interoperability between other devices that do not support CDP. LLDP must be treated in the same manner as CDP and disabled on all interfaces that connect to untrusted networks. In order to accomplish this, issue the **no lldp transmit** and **no lldp receive** interface configuration commands. Issue the **no lldp run** global configuration command in order to disable LLDP globally. LLDP can also be used by a malicious user for reconnaissance and network mapping.

EXEC Timeout

In order to set the interval that the EXEC command interpreter waits for user input before it terminates a session, issue the **exec-timeout** line configuration command. The **exec-timeout** command must be used in order to logout sessions on vty or tty lines that are left idle. By default, sessions are disconnected after 10 minutes of inactivity.

```
!

line con 0
 exec-timeout <minutes> [seconds]
line vty 0 4
 exec-timeout <minutes> [seconds]
!
```

Keepalives for TCP Sessions

The **service tcp-keepalives-in** and **service tcp-keepalives-out** global configuration commands enable a device to send TCP keepalives for TCP sessions. This configuration must be used in order to enable TCP keepalives on inbound connections to the device and outbound connections from the device. 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 Cisco IOS device.

```
!

service tcp-keepalives-in
service tcp-keepalives-out
!
```

Management Interface Use

The management plane of a device is accessed in-band or out-of-band on a physical or logical management interface. Ideally, both in-band and out-of-band management access exists for each network device so that the management plane can be accessed during network outages.

One of the most common interfaces that is used for in-band access to a device is the logical loopback interface. Loopback interfaces are always up, whereas physical interfaces can change state, and the interface can potentially not be accessible. It is recommended to add a loopback interface to each device as a management interface and that it be used exclusively for the management plane. This allows the administrator to apply policies throughout the network for the management plane. Once the loopback interface is configured on a device, it can be used by management plane protocols, such as SSH, SNMP, and syslog, in order to send and receive traffic.

```
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.0
!
```

Memory Threshold Notifications

The feature Memory Threshold Notification, added in Cisco IOS Software Release 12.3(4)T, allows you to mitigate low-memory conditions on a device. This feature uses two methods in order to accomplish this: Memory Threshold Notification and Memory Reservation.

Memory Threshold Notification generates a log message in order to indicate that free memory on a device has fallen lower than the configured threshold. This configuration example shows how to enable this feature with the **memory free low-watermark** global configuration command. This enables a device to generate a notification when available free memory falls lower than the specified threshold, and again when available free memory rises to five percent higher than the specified threshold.

```
!

memory free low-watermark processor <threshold>
memory free low-watermark io <threshold>
!
```

Memory Reservation is used so that sufficient memory is available for critical notifications. This configuration example demonstrates how to enable this feature. This ensures that management processes continue to function when the memory of the device is exhausted.

```
!
memory reserve critical <value>!
```

Refer to Memory Threshold Notifications for more information about this feature.

CPU Thresholding Notification

Introduced in Cisco IOS Software Release 12.3(4)T, the CPU Thresholding Notification feature allows you to detect and be notified when the CPU load on a device crosses a configured threshold. When the threshold is crossed, the device generates and sends an SNMP trap message. Two CPU utilization thresholding methods are supported on Cisco IOS software: Rising Threshold and Falling Threshold.

This example configuration shows how to enable the Rising and Falling Thresholds that trigger a CPU threshold notification message:

```
!

snmp-server enable traps cpu threshold
!

snmp-server host <host-address> <community-string> cpu
!

process cpu threshold type <type> rising <percentage> interval <seconds>
     [falling <percentage> interval <seconds>]
process cpu statistics limit entry-percentage <number> [size <seconds>]
!
```

Refer to CPU Thresholding Notification for more information about this feature.

Reserve Memory for Console Access

In Cisco IOS Software Release 12.4(15)T and later, the Reserve Memory for Console Access feature can be used in order to reserve enough memory to ensure console access to a Cisco IOS device for administrative and troubleshooting purposes. This feature is especially beneficial when the device runs low on memory. You can issue the **memory reserve console** global configuration command in order to enable this feature. This example configures a Cisco IOS device to reserve 4096 kilobytes for this purpose.

```
!
memory reserve console 4096
!
```

Refer to Reserve Memory for Console Access for more information about this feature.

Memory Leak Detector

Introduced in Cisco IOS Software Release 12.3(8)T1, the Memory Leak Detector feature allows you to detect memory leaks on a device. Memory Leak Detector is able to find leaks in all memory pools, packet buffers, and chunks. Memory leaks are static or dynamic allocations of memory that do not serve any useful purpose. This feature focuses on memory allocations that are dynamic. You can use the **show memory debug leaks** EXEC command in order to detect if a memory leak exists.

Buffer Overflow: Detection and Correction of Redzone Corruption

In Cisco IOS Software Release 12.3(7)T and later, the Buffer Overflow: Detection and Correction of Redzone Corruption feature can be enabled by on a device in order to detect and correct a memory block overflow and to continue operations.

These global configuration commands can be used in order to enable this feature. Once configured, the **show memory overflow** command can be used in order to display the buffer overflow detection and correction statistics.

```
!
exception memory ignore overflow io
exception memory ignore overflow processor
!
```

Enhanced Crashinfo File Collection

The Enhanced Crashinfo File Collection feature automatically deletes old crashinfo files. This feature, added in Cisco IOS Software Release 12.3(11)T, allows a device to reclaim space in order to create new crashinfo files when the device crashes. This feature also allows configuration of the number of crashinfo files to be saved.

```
!
exception crashinfo maximum files <number-of-files>
!
```

Network Time Protocol

The Network Time Protocol (NTP) is not an especially dangerous service, but any unneeded service can represent an attack vector. If NTP is used, it is important to explicitly configure a trusted time source and to use proper authentication. Accurate and reliable time is required for syslog purposes, such as during forensic investigations of potential attacks, as well as for successful VPN connectivity when depending on certificates for Phase 1 authentication.

- **NTP Time Zone** - When you configure NTP, the time zone needs to be configured so that timestamps can be accurately correlated. There are usually two approaches to configure the time zone for devices in a network with a global presence. One method is to configure all network devices with the Coordinated Universal Time (UTC) (previously Greenwich Mean Time (GMT)). The other approach is to configure network devices with the local time zone. More information on this feature can be found in "clock timezone" in the Cisco product documentation.
- **NTP Authentication** - If you configure NTP authentication, it provides assurance that NTP messages are exchanged between trusted NTP peers.

Sample configuration using NTP authentication:

Client:

```
(config)#ntp authenticate
(config)#ntp authentication-key 5 md5 ciscotime
(config)#ntp trusted-key 5
(config)#ntp server 172.16.1.5 key 5
```

Server:

```
(config)#ntp authenticate
(config)#ntp authentication-key 5 md5 ciscotime
(config)#ntp trusted-key 5
```

Limit Access to the Network with Infrastructure ACLs

Devised to prevent unauthorized direct communication to network devices, infrastructure access control lists (iACLs) are one of the most critical security controls that can be implemented in networks. Infrastructure ACLs leverage the idea that nearly all network traffic traverses the network and is not destined to the network itself.

An iACL is constructed and applied in order to specify connections from hosts or networks that need to be allowed to network devices. Common examples of these types of connections are eBGP, SSH, and SNMP. After the required connections have been permitted, all other traffic to the infrastructure is explicitly denied. All transit traffic that crosses the network and is not destined to infrastructure devices is then explicitly permitted.

The protections provided by iACLs are relevant to both the management and control planes. The implementation of iACLs can be made easier through the use of distinct addressing for network infrastructure devices. Refer to A Security Oriented Approach to IP Addressing for more information on the security implications of IP addressing.

This example iACL configuration illustrates the structure that must be used as a starting point when you begin the iACL implementation process:

```
!

ip access-list extended ACL-INFRASTRUCTURE-IN
!
!--- Permit required connections for routing protocols and
!--- network management
!

 permit tcp host <trusted-ebgp-peer> host <local-ebgp-address> eq 179
 permit tcp host <trusted-ebgp-peer> eq 179 host <local-ebgp-address>
 permit tcp host <trusted-management-stations> any eq 22
 permit udp host <trusted-netmgmt-servers> any eq 161
!
!--- Deny all other IP traffic to any network device
!

 deny ip any <infrastructure-address-space> <mask>
!
!--- Permit transit traffic
!

 permit ip any any
!
```

Once created, the iACL must be applied to all interfaces that face non-infrastructure devices. This includes interfaces that connect to other organizations, remote access segments, user segments, and segments in data centers.

Refer to Protecting Your Core: Infrastructure Protection Access Control Lists for more information about Infrastructure ACLs.

ICMP Packet Filtering

The Internet Control Message Protocol (ICMP) is designed as an IP control protocol. As such, the messages it conveys can have far-reaching ramifications to the TCP and IP protocols in general. While the network troubleshooting tools **ping** and **traceroute** use ICMP, external ICMP connectivity is rarely needed for the proper operation of a network.

Cisco IOS software provides functionality in order to specifically filter ICMP messages by name or type and code. This example ACL, which must be used with the access control entries (ACEs) from previous examples, allows pings from trusted management stations and NMS servers and blocks all other ICMP packets:

```
!

ip access-list extended ACL-INFRASTRUCTURE-IN
!
!--- Permit ICMP Echo (ping) from trusted management stations and servers
!

 permit icmp host <trusted-management-stations> any echo
 permit icmp host <trusted-netmgmt-servers> any echo
!
!--- Deny all other IP traffic to any network device
!

 deny ip any <infrastructure-address-space> <mask>
!
!--- Permit transit traffic
!

 permit ip any any
!
```

Filter IP Fragments

The filter process for fragmented IP packets can pose a challenge to security devices. This is because the Layer 4 information that is used in order to filter TCP and UDP packets is only present in the initial fragment. Cisco IOS software uses a specific method in order to check non-initial fragments against configured access lists. Cisco IOS software evaluates these non-initial fragments against the ACL and ignores any Layer 4 filtering information. This causes non-initial fragments to be evaluated solely on the Layer 3 portion of any configured ACE.

In this example configuration, if a TCP packet destined to **192.168.1.1** on **port 22** is fragmented in transit, the initial fragment is dropped as expected by the second ACE based on the Layer 4 information within the packet. However, all remaining (non-initial) fragments are allowed by the first ACE based completely on the Layer 3 information in the packet and ACE. This scenario is shown in this configuration:

```
!

ip access-list extended ACL-FRAGMENT-EXAMPLE
 permit tcp any host 192.168.1.1 eq 80
 deny tcp any host 192.168.1.1 eq 22
!>
```

Due to the nonintuitive nature of fragment handling, IP fragments are often inadvertently permitted by ACLs. Fragmentation is also often used in attempts to evade detection by intrusion detection systems. It is for these reasons that IP fragments are often used in attacks, and why they must be explicitly filtered at the top of any configured iACLs. This example ACL includes comprehensive filtering of IP fragments. The functionality from this example must be used in conjunction with the functionality of the previous examples.

```
!

ip access-list extended ACL-INFRASTRUCTURE-IN
!
!--- Deny IP fragments using protocol-specific ACEs to aid in
!--- classification of attack traffic
!

 deny tcp any any fragments
 deny udp any any fragments
 deny icmp any any fragments
 deny ip any any fragments
!
!--- Deny all other IP traffic to any network device
!

 deny ip any <infrastructure-address-space> <mask>
!
!--- Permit transit traffic
!

 permit ip any any
!
```

Refer to Access Control Lists and IP Fragments for more information about how ACL handles fragmented IP packets.

ACL Support for Filtering IP Options

Cisco IOS Software Release 12.3(4)T added support for the use of ACLs to filter IP packets based on the IP options that are contained in the packet. IP options present a security challenge for network devices because these options must be processed as exception packets. This requires a level of CPU effort that is not required for typical packets that traverse the network. The presence of IP options within a packet can also indicate an attempt to subvert security controls in the network or otherwise alter the transit characteristics of a packet. It is for these reasons that packets with IP options must be filtered at the edge of the network.

This example must be used with the ACEs from previous examples in order to include complete filtering of IP packets that contain IP options:

```
!

ip access-list extended ACL-INFRASTRUCTURE-IN
!
!--- Deny IP packets containing IP options
!

 deny ip any any option any-options
!
!--- Deny all other IP traffic to any network device
!

 deny ip any <infrastructure-address-space> <mask>
!
!--- Permit transit traffic
!

 permit ip any any
!
```

ACL Support to Filter on TTL Value

Cisco IOS Software Release 12.4(2)T added ACL support to filter IP packets based on the Time to Live (TTL) value. The TTL value of an IP datagram is decremented by each network device as a packet flows from source to destination. Although initial values vary by operating system, when the TTL of a packet reaches zero, the packet must be dropped. The device that decrements the TTL to zero, and therefore drops the packet, is required in order to generate and send an ICMP Time Exceeded message to the source of the packet.

The generation and transmission of these messages is an exception process. Routers can perform this function when the number of IP packets that are due to expire is low, but if the number of packets due to expire is high, generation and transmission of these messages can consume all available CPU resources. This presents a DoS attack vector. It is for this reason that devices need to be hardened against DoS attacks that utilize a high rate of IP packets that are due to expire.

It is recommended that organizations filter IP packets with low TTL values at the edge of the network. Completely filtering packets with TTL values insufficient to traverse the network mitigates the threat of TTL-based attacks.

This example ACL filters packets with TTL values less than six. This provides protection against TTL expiry attacks for networks up to five hops in width.

```
!

ip access-list extended ACL-INFRASTRUCTURE-IN
!
!--- Deny IP packets with TTL values insufficient to traverse the network
!

 deny ip any any ttl lt 6
!
!--- Deny all other IP traffic to any network device
!

 deny ip any <infrastructure-address-space> <mask>
!
!--- Permit transit traffic
!

 permit ip any any
!
```

Note: Some protocols make legitimate use of packets with low TTL values. eBGP is one such protocol. Refer to TTL Expiry Attack Identification and Mitigation for more information on mitigating TTL expiry-based attacks.

Refer to ACL Support for Filtering on TTL Value for more information about this functionality.

Secure Interactive Management Sessions

Management sessions to devices allow you the ability to view and collect information about a device and its operations. If this information is disclosed to a malicious user, the device can become the target of an attack, compromised, and used in order to perform additional attacks. Anyone with privileged access to a device has the capability for full administrative control of that device. It is imperative to secure management sessions in order to prevent information disclosure and unauthorized access.

Management Plane Protection

In Cisco IOS Software Release 12.4(6)T and later, the feature Management Plane Protection (MPP) allows an administrator to restrict on which interfaces management traffic can be received by a device. This allows the administrator additional control over a device and how the device is accessed.

This example shows how to enable the MPP in order to only allow SSH and HTTPS on the GigabitEthernet0/1 interface:

```
!

control-plane host
  management-interface GigabitEthernet 0/1 allow ssh https
```

```
!
```

Refer to Management Plane Protection for more information about MPP.

Control Plane Protection

Control Plane Protection (CPPr) builds on the functionality of Control Plane Policing in order to restrict and police control plane traffic that is destined to the route processor of the IOS device. CPPr, added in Cisco IOS Software Release 12.4(4)T, divides the control plane into separate control plane categories that are known as subinterfaces. Three control plane subinterfaces exist: Host, Transit and CEF-Exception. In addition, CPPr includes these additional control plane protection features:

- **Port-filtering feature** - This feature provides for the policing or dropping of packets that go to closed or non-listening TCP and UDP ports.
- **Queue-threshold policy feature** - This feature limits the number of packets for a specified protocol that are allowed in the control plane IP input queue.

CPPr allows an administrator to classify, police, and restrict traffic that is sent to a device for management purposes with the host subinterface. Examples of packets that are classified for the host subinterface category include management traffic such as SSH or Telnet and routing protocols.

Note: CPPr does not support IPv6 and is restricted to the IPv4 input path.

Refer to Control Plane Protection Feature Guide - 12.4T and Understanding Control Plane Protection for more information about the Cisco CPPr feature.

Encrypt Management Sessions

Because information can be disclosed in an interactive management session, this traffic must be encrypted so that a malicious user cannot gain access to the data that is transmitted. Traffic encryption allows a secure remote access connection to the device. If the traffic for a management session is sent over the network in cleartext, an attacker can obtain sensitive information about the device and the network.

An administrator is able to establish an encrypted and secure remote access management connection to a device with the SSH or HTTPS (Secure Hypertext Transfer Protocol) features. Cisco IOS software supports SSH Version 1.0 (SSHv1), SSH Version 2.0 (SSHv2), and HTTPS that uses Secure Sockets Layer (SSL) and Transport Layer Security (TLS) for authentication and data encryption. SSHv1 and SSHv2 are not compatible.

Cisco IOS software also supports the Secure Copy Protocol (SCP), which allows an encrypted and secure connection in order to copy device configurations or software images. SCP relies on SSH. This example configuration enables SSH on a Cisco IOS device:

```
!

ip domain-name example.com
!

crypto key generate rsa modulus 2048
!

ip ssh time-out 60
ip ssh authentication-retries 3
ip ssh source-interface GigabitEthernet 0/1
!

line vty 0 4
 transport input ssh
```

```
!
```

This configuration example enables SCP services:

```
!

ip scp server enable
!
```

This is a configuration example for HTTPS services:

```
!

crypto key generate rsa modulus 2048
!

ip http secure-server
!
```

Refer to Configuring Secure Shell on Routers and Switches Running Cisco IOS and Secure Shell (SSH) FAQ for more information about the Cisco IOS software SSH feature.

SSHv2

The SSHv2 support feature introduced in Cisco IOS Software Release 12.3(4)T allows a user to configure SSHv2. (SSHv1 support was implemented in an earlier release of Cisco IOS Software.) SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. The only reliable transport that is defined for SSH is TCP. SSH provides a means to securely access and securely execute commands on another computer or device over a network. The Secure Copy Protocol (SCP) feature that is tunneled over SSH allows for the secure transfer of files.

This example configuration enables SSHv2 (with SSHv1 disabled) on a Cisco IOS device:

```
!

hostname router

!

ip domain-name example.com

!

crypto key generate rsa modulus 2048

!

ip ssh time-out 60
ip ssh authentication-retries 3
ip ssh source-interface GigabitEthernet 0/1

!

ip ssh version 2

!

line vty 0 4
transport input ssh

!
```

Refer to Secure Shell Version 2 Support for more information on the use of SSHv2.

SSHv2 Enhancements for RSA Keys

Cisco IOS SSHv2 supports keyboard-interactive and password-based authentication methods. The SSHv2 Enhancements for RSA Keys feature also supports RSA-based public key authentication for the client and server.

For user authentication, RSA-based user authentication uses a private/public key pair associated with each user for authentication. The user must generate a private/public key pair on the client and configure a public key on the Cisco IOS SSH server in order to complete the authentication.

An SSH user who tries to establish the credentials provides an encrypted signature with the private key. The signature and the user's public key are sent to the SSH server for authentication. The SSH server computes a hash over the public key provided by the user. The hash is used in order to determine if the server has an entry that matches. If a match is found, RSA-based message verification is performed with the public key. Hence, the user is authenticated or denied access based on the encrypted signature.

For server authentication, the Cisco IOS SSH client must assign a host key for each server. When the client tries to establish an SSH session with a server, it receives the signature of the server as part of the key exchange message. If the strict host key checking flag is enabled on the client, the client checks whether it has the host key entry that corresponds to the server preconfigured. If a match is found, the client tries to validate the signature with the server host key. If the server is successfully authenticated, the session establishment continues; otherwise it is terminated and displays a **Server Authentication Failed** message.

This example configuration enables the use of RSA keys with SSHv2 on a Cisco IOS device:

```
!
! Configure a hostname for the device
!

hostname router
!
! Configure a domain name
!

ip domain-name cisco.com
!
! Specify the name of the RSA key pair (in this case, "sshkeys") to use for SSH
!

 ip ssh rsa keypair-name sshkeys
!
! Enable the SSH server for local and remote authentication on the router using
! the "crypto key generate" command
! For SSH version 2, the modulus size must be at least 768 bits
!

 crypto key generate rsa usage-keys label sshkeys modulus 2048
!
! Configure an ssh timeout (in seconds)
!
! The following enables a timeout of 120 seconds for SSH connections
!

ip ssh time-out 120
!
! Configure a limit of five (5) authentication retries
!
```

```
ip ssh authentication-retries 5
!
! Configure SSH version 2
!

ip ssh version 2
!
```

Refer to Secure Shell Version 2 Enhancements for RSA Keys for more information on the use of RSA keys with SSHv2.

This example configuration enables the Cisco IOS SSH server to perform RSA-based user authentication. The user authentication is successful if the RSA public key stored on the server is verified with the public or the private key pair stored on the client.

```
!
! Configure a hostname for the device
!

hostname router
!
! Configure a domain name
!

ip domain-name cisco.com
!
! Generate RSA key pairs using a modulus of 2048 bits
!

crypto key generate rsa modulus 2048
!
! Configure SSH-RSA keys for user and server authentication on the SSH server
!

ip ssh pubkey-chain
!
! Configure the SSH username
!

        username ssh-user
!
! Specify the RSA public key of the remote peer
!
! You must then configure either the key-string command
! (followed by the RSA public key of the remote peer) or the
! key-hash command (followed by the SSH key type and version.)
!
```

Refer to Configuring the Cisco IOS SSH Server to Perform RSA-Based User Authentication for more information on the use of RSA keys with SSHv2.

This example configuration enables the Cisco IOS SSH client to perform RSA-based server authentication.

```
!
!

hostname router
!
ip domain-name cisco.c
!
! Generate RSA key pairs
!
```

```
crypto key generate rsa
!
! Configure SSH-RSA keys for user and server authentication on the SSH server
!

ip ssh pubkey-chain
!
! Enable the SSH server for public-key authentication on the router
!

        server SSH-server-name
!
! Specify the RSA public-key of the remote peer
!
! You must then configure either the key-string command
! (followed by the RSA public key of the remote peer) or the
! key-hash <key-type> <key-name> command (followed by the SSH key
! type and version.)
!
! Ensure that server authentication takes place - The connection will be
! terminated on a failure
!

ip ssh stricthostkeycheck
!
```

Refer to Configuring the Cisco IOS SSH Client to Perform RSA-Based Server Authentication for more information on the use of RSA keys with SSHv2.

Console and AUX Ports

In Cisco IOS devices, console and auxiliary (AUX) ports are asynchronous lines that can be used for local and remote access to a device. You must be aware that console ports on Cisco IOS devices have special privileges. In particular, these privileges allow an administrator to perform the password recovery procedure. In order to perform password recovery, an unauthenticated attacker would need to have access to the console port and the ability to interrupt power to the device or to cause the device to crash.

Any method used in order to access the console port of a device must be secured in a manner that is equal to the security that is enforced for privileged access to a device. Methods used in order to secure access must include the use of AAA, exec-timeout, and modem passwords if a modem is attached to the console.

If password recovery is not required, then an administrator can remove the ability to perform the password recovery procedure using the **no service password-recovery** global configuration command; however, once the **no service password-recovery** command has been enabled, an administrator can no longer perform password recovery on a device.

In most situations, the AUX port of a device must be disabled in order to prevent unauthorized access. An AUX port can be disabled with these commands:

```
!

line aux 0
 transport input none
 transport output none
 no exec
 exec-timeout 0 1
 no password
!
```

Control vty and tty Lines

Interactive management sessions in Cisco IOS software use a tty or virtual tty (vty). A tty is a local asynchronous line to which a terminal can be attached for local access to the device or to a modem for dialup access to a device. Note that ttys can be used for connections to console ports of other devices. This function allows a device with tty lines to act as a console server where connections can be established across the network to the console ports of devices connected to the tty lines. The tty lines for these reverse connections over the network must also be controlled.

A vty line is used for all other remote network connections supported by the device, regardless of protocol (SSH, SCP, or Telnet are examples). In order to ensure that a device can be accessed via a local or remote management session, proper controls must be enforced on both vty and tty lines. Cisco IOS devices have a limited number of vty lines; the number of lines available can be determined with the show line EXEC command. When all vty lines are in use, new management sessions cannot be established, which creates a DoS condition for access to the device.

The simplest form of access control to a vty or tty of a device is through the use of authentication on all lines regardless of the device location within the network. This is critical for vty lines because they are accessible via the network. A tty line that is connected to a modem that is used for remote access to the device, or a tty line that is connected to the console port of other devices are also accessible via the network. Other forms of vty and tty access controls can be enforced with the **transport input** or **access-class** configuration commands, with the use of the CoPP and CPPr features, or if you apply access lists to interfaces on the device.

Authentication can be enforced through the use of AAA, which is the recommended method for authenticated access to a device, with the use of the local user database, or by simple password authentication configured directly on the vty or tty line.

The **exec-timeout** command must be used in order to logout sessions on vty or tty lines that are left idle. The **service tcp-keepalives-in** command must also be used in order to enable TCP keepalives on incoming connections to the device. 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 IOS device.

Control Transport for vty and tty Lines

A vty and tty should be configured in order to accept only encrypted and secure remote access management connections to the device or through the device if it is used as a console server. This section addresses ttys because such lines can be connected to console ports on other devices, which allow the tty to be accessible over the network. In an effort to prevent information disclosure or unauthorized access to the data that is transmitted between the administrator and the device, **transport input ssh** should be used instead of clear-text protocols, such as Telnet and rlogin. The **transport input none** configuration can be enabled on a tty, which in effect disables the use of the tty line for reverse-console connections.

Both vty and tty lines allow an administrator to connect to other devices. In order to limit the type of transport that an administrator can use for outgoing connections, use the **transport output** line configuration command. If outgoing connections are not needed, then **transport output none** should be used. However, if outgoing connections are allowed, then an encrypted and secure remote access method for the connection should be enforced through the use of **transport output ssh**.

Note: IPSec can be used for encrypted and secure remote access connections to a device, if supported. If you use IPSec, it also adds additional CPU overhead to the device. However, SSH must still be enforced as the transport even when IPSec is used.

Warning Banners

In some legal jurisdictions, it can be impossible to prosecute and illegal to monitor malicious users unless they have been notified that they are not permitted to use the system. One method to provide this notification is to place this information into a banner message that is configured with the Cisco IOS software banner login command.

Legal notification requirements are complex, vary by jurisdiction and situation, and should be discussed with legal counsel. Even within jurisdictions, legal opinions can differ. In cooperation with counsel, a banner can provide some or all of the this information:

- Notice that the system is to be logged into or used only by specifically authorized personnel and perhaps information about who can authorize use.
- Notice that any unauthorized use of the system is unlawful and can be subject to civil and criminal penalties.
- Notice that any use of the system can be logged or monitored without further notice and that the resulting logs can be used as evidence in court.
- Specific notices required by local laws.

From a security point of view, rather than legal, a login banner should not contain any specific information about the router name, model, software, or ownership. This information can be abused by malicious users.

Authentication, Authorization, and Accounting

The Authentication, Authorization, and Accounting (AAA) framework is critical in order to secure interactive access to network devices. The AAA framework provides a highly configurable environment that can be tailored based on the needs of the network.

TACACS+ Authentication

TACACS+ is an authentication protocol that Cisco IOS devices can use for authentication of management users against a remote AAA server. These management users can access the IOS device via SSH, HTTPS, telnet, or HTTP.

TACACS+ authentication, or more generally AAA authentication, provides the ability to use individual user accounts for each network administrator. When you do not depend on a single shared password, the security of the network is improved and your accountability is strengthened.

RADIUS is a protocol similar in purpose to TACACS+; however, it only encrypts the password sent across the network. In contrast, TACACS+ encrypts the entire TCP payload, which includes both the username and password. For this reason, TACACS+ should be used in preference to RADIUS when TACACS+ is supported by the AAA server. Refer to TACACS+ and RADIUS Comparison for a more detailed comparison of these two protocols.

TACACS+ authentication can be enabled on a Cisco IOS device with a configuration similar to this example:

```
!

aaa new-model
aaa authentication login default group tacacs+
!

tacacs-server host <ip-address-of-tacacs-server>
tacacs-server key <key>
!
```

The previous configuration can be used as a starting point for an organization-specific AAA authentication template. Refer to Authentication, Authorization, and Accounting for more information about the configuration of AAA.

A method list is a sequential list that describes the authentication methods to be queried in order to authenticate a user. Method lists enable you to designate one or more security protocols to be used for authentication, and thus ensure a backup system for authentication in case the initial method fails. Cisco IOS software uses the first listed method that successfully accepts or rejects a user. Subsequent methods are only attempted in cases where earlier methods fail due to server unavailability or incorrect configuration. Refer to Named Method Lists for Authentication for more information about the configuration of Named Method Lists.

Authentication Fallback

If all configured TACACS+ servers become unavailable, then a Cisco IOS device can rely on secondary authentication protocols. Typical configurations include the use of local or enable authentication if all configured TACACS+ servers are unavailable.

The complete list of options for on-device authentication includes enable, local, and line. Each of these options has advantages. The use of the enable secret is preferred because the secret is hashed with a one-way algorithm that is inherently more secure than the encryption algorithm that is used with the Type 7 passwords for line or local authentication.

However, on Cisco IOS software releases that support the use of secret passwords for locally defined users, fallback to local authentication can be desirable. This allows for a locally defined user to be created for one or more network administrators. If TACACS+ were to become completely unavailable, each administrator can use their local username and password. Although this action does enhance the accountability of network administrators in TACACS+ outages, it significantly increases the administrative burden because local user accounts on all network devices must be maintained.

This configuration example builds upon the previous TACACS+ authentication example in order to include fallback authentication to the password that is configured locally with the **enable secret** command:

```
!

enable secret <password>
!

aaa new-model
aaa authentication login default group tacacs+ enable
!

tacacs-server host <ip-address-of-tacacs-server>
tacacs-server key <key>
!
```

Refer to Configuring Authentication for more information on the use of fallback authentication with AAA.

Use of Type 7 Passwords

Originally designed in order to allow quick decryption of stored passwords, Type 7 passwords are not a secure form of password storage. There are many tools available that can easily decrypt these passwords. The use of Type 7 passwords should be avoided unless required by a feature that is in use on the Cisco IOS device.

The removal of passwords of this type can be facilitated through AAA authentication and the use of the Enhanced Password Security feature, which allows secret passwords to be used with users that are locally defined via the **username** global configuration command. If you cannot fully prevent the use of Type 7 passwords, consider these passwords obfuscated, not encrypted.

See the General Management Plane Hardening section of this document for more information about the removal of Type 7 passwords.

TACACS+ Command Authorization

Command authorization with TACACS+ and AAA provides a mechanism that permits or denies each command that is entered by an administrative user. When the user enters EXEC commands, Cisco IOS sends each command to the configured AAA server. The AAA server then uses its configured policies in order to permit or deny the command for that particular user.

This configuration can be added to the previous AAA authentication example in order to implement command authorization:

```
!

aaa authorization exec default group tacacs none
aaa authorization commands 0 default group tacacs none
aaa authorization commands 1 default group tacacs none
aaa authorization commands 15 default group tacacs none
!
```

Refer to Configuring Authorization for more information about command authorization.

TACACS+ Command Accounting

When configured, AAA command accounting sends information about each EXEC command that is entered to the configured TACACS+ servers. The information sent to the TACACS+ server includes the command executed, the date it was executed, and the username of the user who enters the command. Command accounting is not supported with RADIUS.

This example configuration enables AAA command accounting for EXEC commands entered at privilege levels zero, one, and 15. This configuration builds upon previous examples that include configuration of the TACACS servers.

```
!

aaa accounting exec default start-stop group tacacs
aaa accounting commands 0 default start-stop group tacacs
aaa accounting commands 1 default start-stop group tacacs
aaa accounting commands 15 default start-stop group tacacs
!
```

Refer to Configuring Accounting for more information about the configuration of AAA accounting.

Redundant AAA Servers

The AAA servers that are leveraged in an environment should be redundant and deployed in a fault-tolerant manner. This helps ensure that interactive management access, such as SSH, is possible if an AAA server is unavailable.

When you design or implement a redundant AAA server solution, remember these considerations:

- Availability of AAA servers during potential network failures
- Geographically dispersed placement of AAA servers

- Load on individual AAA servers in steady-state and failure conditions
- Network latency between Network Access Servers and AAA servers
- AAA server databases synchronization

Refer to Deploy the Access Control Servers for more information.

Fortify the Simple Network Management Protocol

This section highlights several methods that can be used in order to secure the deployment of SNMP within IOS devices. It is critical that SNMP be properly secured in order to protect the confidentiality, integrity, and availability of both the network data and the network devices through which this data transits. SNMP provides you with a wealth of information on the health of network devices. This information should be protected from malicious users that want to leverage this data in order to perform attacks against the network.

SNMP Community Strings

Community strings are passwords that are applied to an IOS device to restrict access, both read-only and read-write access, to the SNMP data on the device. These community strings, as with all passwords, should be carefully chosen to ensure they are not trivial. Community strings should be changed at regular intervals and in accordance with network security policies. For example, the strings should be changed when a network administrator changes roles or leaves the company.

These configuration lines configure a read-only community string of READONLY and a read-write community string of READWRITE:

```
!

snmp-server community READONLY RO
snmp-server community READWRITE RW
!
```

Note: The previous community string examples have been chosen in order to clearly explain the use of these strings. For production environments, community strings should be chosen with caution and should consist of a series of alphabetical, numerical, and non-alphanumeric symbols. Refer to Recommendations for Creating Strong Passwords for more information on the selection of non-trivial passwords.

Refer to IOS SNMP Command Reference for more information about this feature.

SNMP Community Strings with ACLs

In addition to the community string, an ACL should be applied that further restricts SNMP access to a select group of source IP addresses. This configuration restricts SNMP read-only access to end host devices that reside in the 192.168.100.0/24 address space and restricts SNMP read-write access to only the end host device at 192.168.100.1.

Note: The devices that are permitted by these ACLs require the proper community string in order to access the requested SNMP information.

```
!

access-list 98 permit 192.168.100.0 0.0.0.255
access-list 99 permit 192.168.100.1
!

snmp-server community READONLY RO 98
snmp-server community READWRITE RW 99
```

```
!
```

Refer to snmp-server community in the Cisco IOS Network Management Command Reference for more information about this feature.

Infrastructure ACLs

Infrastructure ACLs (iACLs) can be deployed in order to ensure that only end hosts with trusted IP addresses can send SNMP traffic to an IOS device. An iACL should contain a policy that denies unauthorized SNMP packets on UDP port 161.

See the Limiting Access to the Network with Infrastructure ACLs section of this document for more information on the use of iACLs.

SNMP Views

SNMP Views are a security feature that can permit or deny access to certain SNMP MIBs. Once a view is created and applied to a community string with the **snmp-server community** community-string view global configuration commands, if you access MIB data, you are restricted to the permissions that are defined by the view. When appropriate, you are advised to use views to limit users of SNMP to the data that they require.

This configuration example restricts SNMP access with the community string LIMITED to the MIB data that is located in the system group:

```
!

snmp-server view VIEW-SYSTEM-ONLY system include
!

snmp-server community LIMITED view VIEW-SYSTEM-ONLY RO
!
```

Refer to Configuring SNMP Support for more information.

SNMP Version 3

SNMP Version 3 (SNMPv3) is defined by RFC3410, RFC3411, RFC3412, RFC3413, RFC3414, and RFC3415 and is an interoperable standards-based protocol for network management. SNMPv3 provides secure access to devices because it authenticates and optionally encrypts packets over the network. Where supported, SNMPv3 can be used in order to add another layer of security when you deploy SNMP. SNMPv3 consists of three primary configuration options:

- **no auth** - This mode does not require any authentication nor any encryption of SNMP packets
- **auth** - This mode requires authentication of the SNMP packet without encryption
- **priv** - This mode requires both authentication and encryption (privacy) of each SNMP packet

An authoritative engine ID must exist in order to use the SNMPv3 security mechanisms - authentication or authentication and encryption - to handle SNMP packets; by default, the engine ID is generated locally. The engine ID can be displayed with the **show snmp engineID** command as shown in this example:

```
router#show snmp engineID
   Local SNMP engineID: 80000009030000152BD35496
   Remote Engine ID          IP-addr    Port
```

Note: If the engineID is changed, all SNMP user accounts must be reconfigured.

The next step is to configure an SNMPv3 group. This command configures a Cisco IOS device for SNMPv3 with an SNMP server group AUTHGROUP and enables only authentication for this group with the **auth** keyword:

```
!
snmp-server group AUTHGROUP v3 auth
!
```

This command configures a Cisco IOS device for SNMPv3 with an SNMP server group PRIVGROUP and enables both authentication and encryption for this group with the **priv** keyword:

```
!
snmp-server group PRIVGROUP v3 priv
!
```

This command configures an SNMPv3 user snmpv3user with an MD5 authentication password of **authpassword** and a 3DES encryption password of **privpassword**:

```
!

snmp-server user snmpv3user PRIVGROUP v3 auth md5 authpassword priv 3des
     privpassword
!
```

Note that **snmp-server user** configuration commands are not displayed in the configuration output of the device as required by RFC 3414; therefore, the user password is not viewable from the configuration. In order to view the configured users, enter the **show snmp user** command as shown in this example:

```
router#show snmp user
User name: snmpv3user
Engine ID: 80000009030000152BD35496
storage-type: nonvolatile        active
Authentication Protocol: MD5
Privacy Protocol: 3DES
Group-name: PRIVGROUP
```

Refer to Configuring SNMP Support for more information about this feature.

Management Plane Protection

The Management Plane Protection (MPP) feature in Cisco IOS software can be used in order to help secure SNMP because it restricts the interfaces through which SNMP traffic can terminate on the device. The MPP feature allows an administrator to designate one or more interfaces as management interfaces. Management traffic is permitted to enter a device only through these management interfaces. After MPP is enabled, no interfaces except designated management interfaces accept network management traffic that is destined to the device.

Note that the MPP is a subset of the CPPr feature and requires a version of IOS that supports CPPr. Refer to Understanding Control Plane Protection for more information on CPPr.

In this example, MPP is used in order to restrict SNMP and SSH access to only the FastEthernet 0/0 interface:

```
!
control-plane host
 management-interface FastEthernet0/0 allow ssh snmp
!
```

Refer to Management Plane Protection Feature Guide for more information.

Logging Best Practices

Event logging provides you visibility into the operation of a Cisco IOS device and the network into which it is deployed. Cisco IOS software provides several flexible logging options that can help achieve the network management and visibility goals of an organization.

These sections provide some basic logging best practices that can help an administrator leverage logging successfully while minimizing the impact of logging on a Cisco IOS device.

Send Logs to a Central Location

You are advised to send logging information to a remote syslog server. This makes it possible to correlate and audit network and security events across network devices more effectively. Note that syslog messages are transmitted unreliably by UDP and in cleartext. For this reason, any protections that a network affords to management traffic (for example, encryption or out-of-band access) should be extended in order to include syslog traffic.

This configuration example configures a Cisco IOS device in order to send logging information to a remote syslog server:

```
!
logging host <ip-address>
!
```

Refer to Identifying Incidents Using Firewall and IOS Router Syslog Events for more information on log correlation.

Integrated in 12.4(15)T and originally introduced in 12.0(26)S, the Logging to Local Nonvolatile Storage (ATA Disk) feature enables system logging messages to be saved on an advanced technology attachment (ATA) flash disk. Messages saved on an ATA drive persist after a router is rebooted.

This configuration lines configure 134,217,728 bytes (128 MB) of logging messages to the syslog directory of the ATA flash (disk0), specifying a file size of 16,384 bytes:

```
logging buffered
logging persistent url disk0:/syslog size 134217728 filesize 16384
```

Before logging messages are written to a file on the ATA disk, Cisco IOS Software checks if there is sufficient disk space. If not, the oldest file of logging messages (by timestamp) is deleted, and the current file is saved. The filename format is log_month:day:year::time.

Note: An ATA flash drive has limited disk space and thus needs to be maintained to avoid overwriting stored data.

This example shows how to copy logging messages from the router ATA flash disk to an external disk on FTP server 192.168.1.129 as part of maintenance procedures:

```
copy disk0:/syslog ftp://myuser/mypass@192.168.1.129/syslog
```

Refer to Logging to Local Nonvolatile Storage (ATA Disk) for more information about this feature.

Logging Level

Each log message that is generated by a Cisco IOS device is assigned one of eight severities that range from level 0, Emergencies, through level 7, Debug. Unless specifically required, you are advised to avoid logging at level 7. Logging at level 7 produces an elevated CPU load on the device that can lead to device and network instability.

The global configuration command **logging trap** level is used in order to specify which logging messages are sent to remote syslog servers. The level specified indicates the lowest severity message that is sent. For buffered logging, the **logging buffered** level command is used.

This configuration example limits log messages that are sent to remote syslog servers and the local log buffer to severities 6 (informational) through 0 (emergencies):

```
!

logging trap 6
logging buffered 6
!
```

Refer to Troubleshooting, Fault Management, and Logging for more information.

Do Not Log to Console or Monitor Sessions

With Cisco IOS software, it is possible to send log messages to monitor sessions - monitor sessions are interactive management sessions in which the EXEC command **terminal monitor** has been issued - and to the console. However, this can elevate the CPU load of an IOS device and therefore is not recommended. Instead, you are advised to send logging information to the local log buffer, which can be viewed with the **show logging** command.

Use the global configuration commands **no logging console** and **no logging monitor** in order to disable logging to the console and monitor sessions. This configuration example shows the use of these commands:

```
!

no logging console
no logging monitor
!
```

Refer to Cisco IOS Network Management Command Reference for more information about global configuration commands.

Use Buffered Logging

Cisco IOS software supports the use of a local log buffer so that an administrator can view locally generated log messages. The use of buffered logging is highly recommended versus logging to either the console or monitor sessions.

There are two configuration options that are relevant when configuring buffered logging: the logging buffer size and the message severities that is stored in the buffer. The size of the **logging buffer** is configured with the global configuration command **logging buffered** size. The lowest severity included in the buffer is configured with the logging buffered severity command. An administrator is able to view the contents of the logging buffer through the **show logging** EXEC command.

This configuration example includes the configuration of a logging buffer of 16384 bytes, as well as a severity of 6, informational, which indicates that messages at levels 0 (emergencies) through 6 (informational) is stored:

```
!

logging buffered 16384 6
```

```
!
```

Refer to Cisco IOS Network Management Command Reference for more information about buffered logging.

Configure Logging Source Interface

In order to provide an increased level of consistency when you collect and review log messages, you are advised to statically configure a logging source interface. Accomplished via the **logging source-interface** interface command, statically configuring a logging source interface ensures that the same IP address appears in all logging messages that are sent from an individual Cisco IOS device. For added stability, you are advised to use a loopback interface as the logging source.

This configuration example illustrates the use of the **logging source-interface** interface global configuration command in order to specify that the IP address of the loopback 0 interface be used for all log messages:

```
!
logging source-interface Loopback 0
!
```

Refer to the Cisco IOS Command Reference for more information.

Configure Logging Timestamps

The configuration of logging timestamps helps you correlate events across network devices. It is important to implement a correct and consistent logging timestamp configuration to ensure that you are able to correlate logging data. Logging timestamps should be configured to include the date and time with millisecond precision and to include the time zone in use on the device.

This example includes the configuration of logging timestamps with millisecond precision within the Coordinated Universal Time (UTC) zone:

```
!
service timestamps log datetime msec show-timezone
!
```

If you prefer not to log times relative to UTC, you can configure a specific local time zone and configure that information to be present in generated log messages. This example shows a device configuration for the Pacific Standard Time (PST) zone:

```
!

clock timezone PST -8
service timestamps log datetime msec localtime show-timezone
!
```

Cisco IOS Software Configuration Management

Cisco IOS software includes several features that can enable a form of configuration management on a Cisco IOS device. Such features include functionality to archive configurations and to rollback the configuration to a previous version as well as create a detailed configuration change log.

Configuration Replace and Configuration Rollback

In Cisco IOS Software Release 12.3(7)T and later, the Configuration Replace and Configuration Rollback features allow you to archive the Cisco IOS device configuration on the device. Stored manually or automatically, the configurations in this archive can be used in order to replace the current running configuration with the **configure replace** filename command. This is in contrast to the **copy** filename **running-config** command. The **configure replace** filename command replaces the running configuration as opposed to the merge performed by the **copy** command.

You are advised to enable this feature on all Cisco IOS devices in the network. Once enabled, an administrator can cause the current running configuration to be added to the archive with the **archive config** privileged EXEC command. The archived configurations can be viewed with the **show archive** EXEC command.

This example illustrates the configuration of automatic configuration archiving. This example instructs the Cisco IOS device to store archived configurations as files named archived-config-N on the disk0: file system, to maintain a maximum of 14 backups, and to archive once per day (1440 minutes) and when an administrator issues the **write memory** EXEC command.

```
!

archive
 path disk0:archived-config
 maximum 14
 time-period 1440
 write-memory
!
```

Although the configuration archive functionality can store up to 14 backup configurations, you are advised to consider the space requirements before you use the **maximum** command.

Exclusive Configuration Change Access

Added to Cisco IOS Software Release 12.3(14)T, the Exclusive Configuration Change Access feature ensures that only one administrator makes configuration changes to a Cisco IOS device at a given time. This feature helps eliminate the undesirable impact of simultaneous changes made to related configuration components. This feature is configured with the global configuration command **configuration mode exclusive** mode and operates in one of two modes: auto and manual. In auto-mode, the configuration automatically locks when an administrator issues the **configure terminal** EXEC command. In manual mode, the administrator uses the **configure terminal lock** command in order to lock the configuration when it enters configuration mode.

This example illustrates the configuration of this feature for automatic configuration locking:

```
!
configuration mode exclusive auto
!
```

Cisco IOS Software Resilient Configuration

Added in Cisco IOS Software Release 12.3(8)T, the Resilient Configuration feature makes it possible to securely store a copy of the Cisco IOS software image and device configuration that is currently used by a Cisco IOS device. When this feature is enabled, it is not possible to alter or remove these backup files. You are advised to enable this feature in order to prevent both inadvertent and malicious attempts to delete these files.

```
!
secure boot-image
secure boot-config!
```

Once this feature is enabled, it is possible to restore a deleted configuration or Cisco IOS software image. The current running state of this feature can be displayed with the **show secure boot** EXEC command.

Digitally Signed Cisco Software

Added in Cisco IOS Software Release 15.0(1)M for the Cisco 1900, 2900, and 3900 Series routers, the Digitally Signed Cisco Software feature facilitates the use of Cisco IOS Software that is digitally signed and thus trusted, with the use of secure asymmetrical (public-key) cryptography.

A digitally signed image carries an encrypted (with a private key) hash of itself. Upon check, the device decrypts the hash with the corresponding public key from the keys it has in its key store and also calculates its own hash of the image. If the decrypted hash matches the calculated image hash, the image has not been tampered with and can be trusted.

Digitally signed Cisco software keys are identified by the type and version of the key. A key can be a special, production, or rollover key type. Production and special key types have an associated key version that increments alphabetically whenever the key is revoked and replaced. ROMMON and regular Cisco IOS images are both signed with a special or production key when you use the Digitally Signed Cisco Software feature. The ROMMON image is upgradable and must be signed with the same key as the special or production image that is loaded.

This command verifies the integrity of image c3900-universalk9-mz.SSA in flash with the keys in the device key store:

```
show software authenticity file flash0:c3900-universalk9-mz.SSA
```

The Digitally Signed Cisco Software feature was also integrated in Cisco IOS XE Release 3.1.0.SG for the Cisco Catalyst 4500 E-Series Switches.

Refer to Digitally Signed Cisco Software for more information about this feature.

In Cisco IOS Software Release 15.1(1)T and later, Key Replacement for Digitally Signed Cisco Software was introduced. Key replacement and revocation replaces and removes a key that is used for a Digitally Signed Cisco Software check from a platform's key storage. Only special and production keys can be revoked in the event of a key compromise.

A new (special or production) key for a (special or production) image comes in a (production or revocation) image that is used in order to revoke the previous special or production key. The revocation image integrity is verified with a rollover key that comes prestored on the platform. A rollover key does not change. When you revoke a production key, after the revocation image is loaded, the new key it carries is added to the key store and the corresponding old key can be revoked as long as ROMMON image is upgraded and the new production image is booted. When you revoke a special key, a production image is loaded. This image adds the new special key and can revoke the old special key. After you upgrade ROMMON, the new special image can be booted.

This example describes revocation of a special key. These commands add the new special key to the key store from the current production image, copy a new ROMMON image (C3900_rom-monitor.srec.SSB) to the storage area (usbflash0:), upgrade the ROMMON file, and revoke the old special key:

```
software authenticity key add special
copy tftp://192.168.1.129/C3900_rom-monitor.srec.SSB usbflash0:
upgrade rom-monitor file usbflash0:C3900_PRIV_RM2.srec.SSB
software authenticity key revoke special
```

A new special image (c3900-universalk9-mz.SSB) can then be copied to the flash to be loaded and the signature of the image is verified with the newly added special key (.SSB):

```
copy /verify tftp://192.168.1.129/c3900-universalk9-mz.SSB flash:
```

Key revocation and replacement is not supported on Catalyst 4500 E-Series Switches that run Cisco IOS XE Software, although these switches do support the Digitally Signed Cisco Software feature.

Refer to the Digitally Signed Cisco Software Key Revocation and Replacement section of the Digitally Signed Cisco Software guide for more information about this feature.

Configuration Change Notification and Logging

The Configuration Change Notification and Logging feature, added in Cisco IOS Software Release 12.3(4)T, makes it possible to log the configuration changes made to a Cisco IOS device. The log is maintained on the Cisco IOS device and contains the user information of the individual who made the change, the configuration command entered, and the time that the change was made. This functionality is enabled with the **logging enable** configuration change logger configuration mode command. The optional commands **hidekeys** and **logging size** entries are used in order to improve the default configuration becuase they prevent the logging of password data and increase the length of the change log.

You are advised to enable this functionality so that the configuration change history of a Cisco IOS device can be more easily understood. Additionally, you are advised to use the **notify syslog** configuration command in order to enable the generation of syslog messages when a configuration change is made.

```
!

archive
 log config
  logging enable
  logging size 200
  hidekeys
  notify syslog
!
```

After the Configuration Change Notification and Logging feature has been enabled, the privileged EXEC command **show archive log config all** can be used in order to view the configuration log.

Control Plane

Control plane functions consist of the protocols and processes that communicate between network devices in order to move data from source to destination. This includes routing protocols such as the Border Gateway Protocol, as well as protocols like ICMP and the Resource Reservation Protocol (RSVP).

It is important that events in the management and data planes do not adversely affect the control plane. Should a data plane event such as a DoS attack impact the control plane, the entire network can become unstable. This information about Cisco IOS software features and configurations can help ensure the resilience of the control plane.

General Control Plane Hardening

Protection of the control plane of a network device is critical because the control plane ensures that the management and data planes are maintained and operational. If the control plane were to become unstable during a security incident, it can be impossible for you to recover the stability of the network.

In many cases, you can disable the reception and transmission of certain types of messages on an interface in order to minimize the amount of CPU load that is required to process unneeded packets.

IP ICMP Redirects

An ICMP redirect message can be generated by a router when a packet is received and transmitted on the same interface. In this situation, the router forwards the packet and sends an ICMP redirect message back to the sender of the original packet. This behavior allows the sender to bypass the router and forward future packets directly to the destination (or to a router closer to the destination). In a properly functioning IP network, a router sends redirects only to hosts on its own local subnets. In other words, ICMP redirects should never go beyond a Layer 3 boundary.

There are two types of ICMP redirect messages: redirect for a host address and redirect for an entire subnet. A malicious user can exploit the ability of the router to send ICMP redirects by continually sending packets to the router, which forces the router to respond with ICMP redirect messages, and results in an adverse impact on the CPU and performance of the router. In order to prevent the router from sending ICMP redirects, use the **no ip redirects** interface configuration command.

ICMP Unreachables

Filtering with an interface access list elicits the transmission of ICMP unreachable messages back to the source of the filtered traffic. The generation of these messages can increase CPU utilization on the device. In Cisco IOS software, ICMP unreachable generation is limited to one packet every 500 milliseconds by default. ICMP unreachable message generation can be disabled with the interface configuration command **no ip unreachables**. ICMP unreachable rate limiting can be changed from the default with the global configuration command **ip icmp rate-limit unreachable** interval-in-ms.

Proxy ARP

Proxy ARP is the technique in which one device, usually a router, answers ARP requests that are intended for another device. By "faking" its identity, the router accepts responsibility for routing packets to the real destination. Proxy ARP can help machines on a subnet reach remote subnets without configuring routing or a default gateway. Proxy ARP is defined in RFC 1027 .

There are several disadvantages to proxy ARP utilization. It can result in an increase in the amount of ARP traffic on the network segment and resource exhaustion and man-in-the-middle attacks. Proxy ARP presents a resource exhaustion attack vector because each proxied ARP request consumes a small amount of memory. An attacker can be able to exhaust all available memory if it sends a large number of ARP requests.

Man-in-the-middle attacks enable a host on the network to spoof the MAC address of the router, which results in unsuspecting hosts sending traffic to the attacker. Proxy ARP can be disabled with the interface configuration command **no ip proxy-arp**.

Refer to Enabling Proxy ARP for more information on this feature.

Limit CPU Impact of Control Plane Traffic

Protection of the control plane is critical. Because application performance and end-user experience can suffer without the presence of data and management traffic, the survivability of the control plane ensures that the other two planes are maintained and operational.

Understand Control Plane Traffic

In order properly protect the control plane of the Cisco IOS device, it is essential to understand the types of traffic that is process switched by the CPU. Process switched traffic normally consists of two different types of traffic. The first type of traffic is directed to the Cisco IOS device and must be handled directly by the Cisco IOS device CPU. This traffic consists of this category:

- **Receive adjacency traffic** - This traffic contains an entry in the Cisco Express Forwarding (CEF) table whereby the next router hop is the device itself, which is indicated by the term receive in the **show ip cef** CLI output. This indication is the case for any IP address that requires direct handling by the Cisco IOS device CPU, which includes interface IP addresses, multicast address space, and broadcast address space.

The second type of traffic that is handled by the CPU is data plane traffic - traffic with a destination beyond the Cisco IOS device itself - which requires special processing by the CPU. Although not an exhaustive list of CPU impacting data plane traffic, these types of traffic are process switched and can therefore affect the operation of the control plane:

- **Access Control List logging** - ACL logging traffic consists of any packets that are generated due to a match (permit or deny) of an ACE on which the log keyword is used.
- **Unicast Reverse Path Forwarding (Unicast RPF)** - Unicast RPF, used in conjunction with an ACL, can result in the process switching of certain packets.
- **IP Options** - Any IP packets with options included must be processed by the CPU.
- **Fragmentation** - Any IP packet that requires fragmentation must be passed to the CPU for processing.
- **Time-to-live (TTL) Expiry** - Packets which have a TTL value less than or equal to one require Internet Control Message Protocol Time Exceeded (ICMP Type 11, Code 0) messages to be sent, which results in CPU processing.
- **ICMP Unreachables** - Packets that result in ICMP unreachable messages due to routing, MTU, or filtering is processed by the CPU.
- **Traffic Requiring an ARP Request** - Destinations for which an ARP entry does not exist require processing by the CPU.
- **Non-IP Traffic** - All non-IP traffic is processed by the CPU.

This list details several methods to determine which types of traffic are being processed by the Cisco IOS device CPU:

- The **show ip cef** command provides the next-hop information for each IP prefix that is contained in the CEF table. As indicated previously, entries that contain receive as the "Next Hop" are considered receive adjacencies and indicate that traffic must be sent directly to the CPU.
- The **show interface switching** command provides information on the number of packets that are process switched by a device.
- The **show ip traffic** command provides information on the number of IP packets:
    - ♦ with a local destination (that is, receive adjacency traffic)
    - ♦ with options
    - ♦ that require fragmentation
    - ♦ that are sent to broadcast address space
    - ♦ that are sent to multicast address space
- Receive adjacency traffic can be identified through the use of the **show ip cache flow** command. Any flows that are destined to the Cisco IOS device has a Destination Interface (DstIf) of local.

- **Control Plane Policing** can be used in order to identify the type and rate of traffic that reaches the control plane of the Cisco IOS device. Control plane policing can be performed through the use of granular classification ACLs, logging, and the use of the **show policy-map control-plane** command.

Infrastructure ACLs

Infrastructure ACLs (iACLs) limit external communication to the devices of the network. Infrastructure ACLs are extensively covered in the Limit Access to the Network with Infrastructure ACLs section of this document.

You are advised to implement iACLs in order to protect the control plane of all network devices.

Receive ACLs

For distributed platforms, Receive ACLs (rACLs) can be an option for Cisco IOS Software Releases 12.0(21)S2 for the 12000 (GSR), 12.0(24)S for the 7500, and 12.0(31)S for the 10720. The rACL protects the device from harmful traffic before the traffic impacts the route processor. Receive ACLs are designed to only protect the device on which it is configured and transit traffic is not affected by an rACL. As a result, the destination IP address any that is used in the example ACL entries below only refers to the physical or virtual IP addresses of the router. Receive ACLs are also considered a network security best practice and should be considered as a long-term addition to good network security.

This is the receive path ACL that is written to permit SSH (TCP port 22) traffic from trusted hosts on the 192.168.100.0/24 network:

```
!
!--- Permit SSH from trusted hosts allowed to the device.
!

access-list 151 permit tcp 192.168.100.0 0.0.0.255 any eq 22
!
!--- Deny SSH from all other sources to the RP.
!

access-list 151 deny tcp any any eq 22
!
!--- Permit all other traffic to the device.
!--- according to security policy and configurations.
!

access-list 151 permit ip any any
!
!--- Apply this access list to the receive path.
!

ip receive access-list 151
!
```

Refer to GSR: Receive Access Control Lists in order to help identify and allow legitimate traffic to a device and deny all unwanted packets.

Control Plane Policing

The Control Plane Policing (CoPP) feature can also be used in order to restrict IP packets that are destined to the infrastructure device. In this example, only SSH traffic from trusted hosts is permitted to reach the Cisco IOS device CPU.

Note: Dropping traffic from unknown or untrusted IP addresses can prevent hosts with dynamically-assigned IP addresses from connecting to the Cisco IOS device.

```
!

access-list 152 deny tcp <trusted-addresses> <mask> any eq 22
access-list 152 permit tcp any any eq 22
access-list 152 deny ip any any
!

class-map match-all COPP-KNOWN-UNDESIRABLE
 match access-group 152
!

policy-map COPP-INPUT-POLICY
 class COPP-KNOWN-UNDESIRABLE
  drop
!

control-plane
 service-policy input COPP-INPUT-POLICY
!
```

In the previous CoPP example, the ACL entries that match the unauthorized packets with the permit action result in a discard of these packets by the policy-map drop function, while packets that match the deny action are not affected by the policy-map drop function.

CoPP is available in Cisco IOS Software Release trains 12.0S, 12.2SX, 12.2S, 12.3T, 12.4, and 12.4T.

Refer to Deploying Control Plane Policing for more information on the configuration and use of the CoPP feature.

Control Plane Protection

Control Plane Protection (CPPr), introduced in Cisco IOS Software Release 12.4(4)T, can be used in order to restrict or police control plane traffic that is destined to the CPU of the Cisco IOS device. While similar to CoPP, CPPr has the ability to restrict traffic with finer granularity. CPPr divides the aggregate control plane into three separate control plane categories known as subinterfaces. Subinterfaces exist for Host, Transit, and CEF-Exception traffic categories. In addition, CPPr includes these control plane protection features:

- **Port-filtering feature** - This feature provides for policing and dropping of packets that are sent to closed or non-listening TCP or UDP ports.
- **Queue-thresholding feature** - This feature limits the number of packets for a specified protocol that are allowed in the control-plane IP input queue.

Refer to Control Plane Protection and Understanding Control Plane Protection (CPPr) for more information on the configuration and use of the CPPr feature.

Hardware Rate Limiters

The Cisco Catalyst 6500 Series Supervisor Engine 32 and Supervisor Engine 720 support platform-specific, hardware-based rate limiters (HWRLs) for special networking scenarios. These hardware rate limiters are referred to as special-case rate limiters because they cover a specific predefined set of IPv4, IPv6, unicast, and multicast DoS scenarios. HWRLs can protect the Cisco IOS device from a variety of attacks that require packets to be processed by the CPU.

There are several HWRLs that are enabled by default. Refer to PFC3 Hardware-based Rate Limiter Default Settings for more information.

Refer to Hardware-Based Rate Limiters on the PFC3 for more information about HWRLs.

Secure BGP

The Border Gateway Protocol (BGP) is the routing foundation of the Internet. As such, any organization with more than modest connectivity requirements often uses BGP. BGP is often targeted by attackers because of its ubiquity and the "set and forget" nature of BGP configurations in smaller organizations. However, there are many BGP-specific security features that can be leveraged to increase the security of a BGP configuration.

This provides an overview of the most important BGP security features. Where appropriate, configuration recommendations are made.

TTL-based Security Protections

Each IP packet contains a 1-byte field known as the Time to Live (TTL). Each device that an IP packet traverses decrements this value by one. The starting value varies by operating system and typically ranges from 64 to 255. A packet is dropped when its TTL value reaches zero.

Known as both the Generalized TTL-based Security Mechanism (GTSM) and BGP TTL Security Hack (BTSH), a TTL-based security protection leverages the TTL value of IP packets in order to ensure that the BGP packets that are received are from a directly connected peer. This feature often requires coordination from peering routers; however, once enabled, it can completely defeat many TCP-based attacks against BGP.

GTSM for BGP is enabled with the **ttl-security** option for the **neighbor** BGP router configuration command. This example illustrates the configuration of this feature:

```
!

router bgp <asn>
 neighbor <ip-address> remote-as <remote-asn>
 neighbor <ip-address> ttl-security hops <hop-count>
!
```

As BGP packets are received, the TTL value is checked and must be greater than or equal to 255 minus the hop-count specified.

BGP Peer Authentication with MD5

Peer authentication with MD5 creates an MD5 digest of each packet sent as part of a BGP session. Specifically, portions of the IP and TCP headers, TCP payload, and a secret key are used in order to generate the digest.

The created digest is then stored in TCP option Kind 19, which was created specifically for this purpose by RFC 2385 . The receiving BGP speaker uses the same algorithm and secret key in order to regenerate the message digest. If the received and computed digests are not identical, the packet is discarded.

Peer authentication with MD5 is configured with the **password** option to the **neighbor** BGP router configuration command. The use of this command is illustrated as follows:

```
!

router bgp <asn>
 neighbor <ip-address> remote-as <remote-asn>
 neighbor <ip-address> password <secret>
!
```

Refer to Neighbor Router Authentication for more information about BGP peer authentication with MD5.

Configure Maximum Prefixes

BGP prefixes are stored by a router in memory. The more prefixes that a router must hold, the more memory that BGP must consume. In some configurations, a subset of all Internet prefixes can be stored, such as in configurations that leverage only a default route or routes for a provider's customer networks.

In order to prevent memory exhaustion, it is important to configure the maximum number of prefixes that is accepted on a per-peer basis. It is recommended that a limit be configured for each BGP peer.

When you configure this feature with the **neighbor maximum-prefix** BGP router configuration command, one argument is required: the maximum number of prefixes that are accepted before a peer is shutdown. Optionally, a number from 1 to 100 can also be entered. This number represents the percentage of the maximum prefixes value at which point a log message is sent.

```
!

router bgp <asn>
 neighbor <ip-address> remote-as <remote-asn>
 neighbor <ip-address> maximum-prefix <shutdown-threshold> <log-percent>
!
```

Refer to Configuring the BGP Maximum-Prefix Feature for more information about per-peer maximum prefixes.

Filter BGP Prefixes with Prefix Lists

Prefix lists allow a network administrator to permit or deny specific prefixes that are sent or received via BGP. Prefix lists should be used where possible in order to ensure network traffic is sent over the intended paths. Prefix lists should be applied to each eBGP peer in both the inbound and outbound directions.

Configured prefix lists limit the prefixes that are sent or received to those specifically permitted by the routing policy of a network. If this is not feasible due to the large number of prefixes received, a prefix list should be configured to specifically block known bad prefixes. These known bad prefixes include unallocated IP address space and networks that are reserved for internal or testing purposes by RFC 3330. Outbound prefix lists should be configured to specifically permit only the prefixes that an organization intends to advertise.

This configuration example uses prefix lists to limit the routes that are learned and advertised. Specifically, only a default route is allowed inbound by prefix list BGP-PL-INBOUND, and the prefix 192.168.2.0/24 is the only route allowed to be advertised by BGP-PL-OUTBOUND.

```
!

ip prefix-list BGP-PL-INBOUND seq 5 permit 0.0.0.0/0
ip prefix-list BGP-PL-OUTBOUND seq 5 permit 192.168.2.0/24
!

router bgp <asn>
 neighbor <ip-address> prefix-list BGP-PL-INBOUND in
 neighbor <ip-address> prefix-list BGP-PL-OUTBOUND out
!
```

Refer to Connecting to a Service Provider Using External BGP for complete coverage of BGP prefix filtering.

Filter BGP Prefixes with Autonomous System Path Access Lists

BGP autonomous system (AS) path access lists allows the user to filter received and advertised prefixes based on the AS-path attribute of a prefix. This can be used in conjunction with prefix lists in order to establish a robust set of filters.

This configuration example uses AS path access lists in order to restrict inbound prefixes to those originated by the remote AS and outbound prefixes to those originated by the local autonomous system. Prefixes that are sourced from all other autonomous systems are filtered and not installed in the routing table.

```
!

ip as-path access-list 1 permit ^65501$
ip as-path access-list 2 permit ^$
!

router bgp <asn>
 neighbor <ip-address> remote-as 65501
 neighbor <ip-address> filter-list 1 in
 neighbor <ip-address> filter-list 2 out
!
```

Secure Interior Gateway Protocols

The ability of a network to properly forward traffic and recover from topology changes or faults is dependent on an accurate view of the topology. You can often run an Interior Gateway Protocol (IGP) in order provide this view. By default, IGPs are dynamic and discover additional routers that communicate with the particular IGP in use. IGPs also discover routes that can be used during a network link failure.

These subsections provide an overview of the most important IGP security features. Recommendations and examples that cover Routing Information Protocol Version 2 (RIPv2), Enhanced Interior Gateway Routing Protocol (EIGRP), and Open Shortest Path First (OSPF) are provided when appropriate.

Routing Protocol Authentication and Verification with Message Digest 5

Failure to secure the exchange of routing information allows an attacker to introduce false routing information into the network. By using password authentication with routing protocols between routers, you can aid the security of the network. However, because this authentication is sent as cleartext, it can be simple for an attacker to subvert this security control.

By adding MD5 hash capabilities to the authentication process, routing updates no longer contain cleartext passwords, and the entire contents of the routing update is more resistant to tampering. However, MD5 authentication is still susceptible to brute force and dictionary attacks if weak passwords are chosen. You are advised to use passwords with sufficient randomization. Since MD5 authentication is much more secure when compared to password authentication, these examples are specific to MD5 authentication. IPSec can also be used in order to validate and secure routing protocols, but these examples do not detail its use.

EIGRP and RIPv2 utilize Key Chains as part of the configuration. Refer to key for more information on the configuration and use of Key Chains.

This is an example configuration for EIGRP router authentication using MD5:

```
!

key chain <key-name>
 key <key-identifier>
 key-string <password>
```

```
!

interface <interface>
 ip authentication mode eigrp <as-number> md5
 ip authentication key-chain eigrp <as-number> <key-name>
!
```

This is an example MD5 router authentication configuration for RIPv2. RIPv1 does not support authentication.

```
!

key chain <key-name>
 key <key-identifier>
 key-string <password>
!

interface <interface>
 ip rip authentication mode md5
 ip rip authentication key-chain <key-name>
!
```

This is an example configuration for OSPF router authentication using MD5. OSPF does not utilize Key Chains.

```
!

interface <interface>
 ip ospf message-digest-key <key-id> md5 <password>
!

router ospf <process-id>
 network 10.0.0.0 0.255.255.255 area 0
 area 0 authentication message-digest
!
```

Refer to Configuring OSPF for more information.

Passive-Interface Commands

Information leaks, or the introduction of false information into an IGP, can be mitigated through use of the **passive-interface** command that assists in controlling the advertisement of routing information. You are advised not to advertise any information to networks that are outside your administrative control.

This example demonstrates usage of this feature:

```
!

router eigrp <as-number>
 passive-interface default
 no passive-interface <interface>
!
```

Route Filtering

In order to reduce the possibility that you introduce false routing information in the network, you must use Route Filtering. Unlike the **passive-interface** router configuration command, routing occurs on interfaces once route filtering is enabled, but the information that is advertised or processed is limited.

For EIGRP and RIP, usage of the **distribute-list** command with the **out** keyword limits what information is advertised, while usage of the **in** keyword limits what updates are processed. The **distribute-list** command is available for OSPF, but it does not prevent a router from propagating filtered routes. Instead, the **area filter-list** command can be used.

This EIGRP example filters outbound advertisements with the **distribute-list** command and a prefix list:

```
!

ip prefix-list <list-name> seq 10 permit <prefix>
!

router eigrp <as-number>
 passive-interface default
 no passive-interface <interface>
 distribute-list prefix <list-name> out <interface>
!
```

This EIGRP example filters inbound updates with a prefix list:

```
!

ip prefix-list <list-name> seq 10 permit <prefix>
!

router eigrp <as-number>
 passive-interface default
 no passive-interface <interface>
 distribute-list prefix <list-name> in <interface>
!
```

Refer to Configuring IP Routing Protocol-Independent Features for more information about how to control the advertising and processing of routing updates.

This OSPF example uses a prefix list with the OSPF-specific **area filter-list** command:

```
!

ip prefix-list <list-name> seq 10 permit <prefix>
!

router ospf <process-id>
 area <area-id> filter-list prefix <list-name> in
!
```

Routing Process Resource Consumption

Routing Protocol prefixes are stored by a router in memory, and resource consumption increases with additional prefixes that a router must hold. In order to prevent resource exhaustion, it is important to configure the routing protocol to limit resource consumption. This is possible with OSPF if you use the Link State Database Overload Protection feature.

This example demonstrates configuration of the OSPF Link State Database Overload Protection feature:

```
!

router ospf <process-id>
 max-lsa <maximum-number>
!
```

Refer to Limiting the Number of Self-Generating LSAs for an OSPF Process for more information on OSPF Link State Database Overload Protection.

Secure First Hop Redundancy Protocols

First Hop Redundancy Protocols (FHRPs) provide resiliency and redundancy for devices that act as default gateways. This situation and these protocols are commonplace in environments where a pair of Layer 3 devices provides default gateway functionality for a network segment or set of VLANs that contain servers or workstations.

The Gateway Load-Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), and Virtual Router Redundancy Protocol (VRRP) are all FHRPs. By default, these protocols communicate with unauthenticated communications. This kind of communication can allow an attacker to pose as an FHRP-speaking device to assume the default gateway role on the network. This takeover would allow an attacker to perform a man-in-the-middle attack and intercept all user traffic that exits the network.

In order to prevent this type of attack, all FHRPs that are supported by Cisco IOS software include an authentication capability with either MD5 or text strings. Because of the threat posed by unauthenticated FHRPs, it is recommended that instances of these protocols use MD5 authentication. This configuration example demonstrates the use of GLBP, HSRP, and VRRP MD5 authentication:

```
!

interface FastEthernet 1
 description *** GLBP Authentication ***
 glbp 1 authentication md5 key-string <glbp-secret>
 glbp 1 ip 10.1.1.1
!

interface FastEthernet 2
 description *** HSRP Authentication ***
 standby 1 authentication md5 key-string <hsrp-secret>
 standby 1 ip 10.2.2.1
!

interface FastEthernet 3
 description *** VRRP Authentication ***
 vrrp 1 authentication md5 key-string <vrrp-secret>
 vrrp 1 ip 10.3.3.1
!
```

Data Plane

Although the data plane is responsible for moving data from source to destination, within the context of security, the data plane is the least important of the three planes. It is for this reason that it is important to protect the management and control planes in preference over the data plane when you secure a network device .

However, within the data plane itself, there are many features and configuration options that can help secure traffic. These sections detail these features and options such that you can more easily secure your network.

General Data Plane Hardening

The vast majority of data plane traffic flows across the network as determined by the network's routing configuration. However, IP network functionality exists to alter the path of packets across the network. Features such as IP Options, specifically the source routing option, form a security challenge in today's networks.

The use of Transit ACLs is also relevant to the hardening of the data plane. See the Filter Transit Traffic with Transit ACLs section of this document for more information.

IP Options Selective Drop

There are two security concerns presented by IP options. Traffic that contains IP options must be process-switched by Cisco IOS devices, which can lead to elevated CPU load. IP options also include the functionality to alter the path that traffic takes through the network, which potentially allows it to subvert security controls.

Due to these concerns, the global configuration command **ip options {drop | ignore}** has been added to Cisco IOS Software Releases 12.3(4)T, 12.0(22)S, and 12.2(25)S. In the first form of this command, **ip options drop**, all IP packets that contain IP options that are received by the Cisco IOS device are dropped. This prevents both the elevated CPU load and possible subversion of security controls that IP options can enable.

The second form of this command, **ip options ignore**, configures the Cisco IOS device to ignore IP options that are contained in received packets. While this does mitigate the threats related to IP options for the local device, it is possible that downstream devices could be affected by the presence of IP options. It is for this reason that the **drop** form of this command is highly recommended. This is demonstrated in the configuration example:

```
!
ip options drop
!
```

Note that some protocols, for example the RSVP, make legitimate use of IP options. The functionality of these protocols is impacted by this command.

Once IP Options Selective Drop has been enabled, the **show ip traffic** EXEC command can be used in order to determine the number of packets that are dropped due to the presence of IP options. This information is present in the forced drop counter.

Refer to ACL IP Options Selective Drop for more information about this feature.

Disable IP Source Routing

IP source routing leverages the Loose Source Route and Record Route options in tandem or the Strict Source Route along with the Record Route option to enable the source of the IP datagram to specify the network path a packet takes. This functionality can be used in attempts to route traffic around security controls in the network.

If IP options have not been completely disabled via the IP Options Selective Drop feature, it is important that IP source routing is disabled. IP source routing, which is enabled by default in all Cisco IOS Software Releases, is disabled via the **no ip source-route** global configuration command. This configuration example illustrates the use of this command:

```
!
no ip source-route
!
```

Disable ICMP Redirects

ICMP redirects are used in order to inform a network device of a better path to an IP destination. By default, the Cisco IOS software sends a redirect if it receives a packet that must be routed through the interface it was received.

In some situations, it might be possible for an attacker to cause the Cisco IOS device to send many ICMP redirect messages, which results in an elevated CPU load. For this reason, it is recommended that the transmission of ICMP redirects be disabled. ICMP redirects are disabled with the interface configuration **no ip redirects** command , as shown in the example configuration:

```
!

interface FastEthernet 0
 no ip redirects
!
```

Disable or Limit IP Directed Broadcasts

IP Directed Broadcasts make it possible to send an IP broadcast packet to a remote IP subnet. Once it reaches the remote network, the forwarding IP device sends the packet as a Layer 2 broadcast to all stations on the subnet. This directed broadcast functionality has been leveraged as an amplification and reflection aid in several attacks, including the smurf attack.

Current versions of Cisco IOS software have this functionality disabled by default; however, it can be enabled via the **ip directed-broadcast** interface configuration command. Releases of Cisco IOS software prior to 12.0 have this functionality enabled by default.

If a network absolutely requires directed broadcast functionality, its use should be controlled. This is possible with the use of an access control list as an option to the **ip directed-broadcast** command. This configuration example limits directed broadcasts to those UDP packets that originate at a trusted network, 192.168.1.0/24:

```
!

access-list 100 permit udp 192.168.1.0 0.0.0.255 any
!

interface FastEthernet 0
 ip directed-broadcast 100
!
```

Filter Transit Traffic with Transit ACLs

It is possible to control what traffic transits the network with the use of transit ACLs (tACLs). This is in contrast to infrastructure ACLs that seek to filter traffic that is destined to the network itself. The filtering provided by tACLs is beneficial when it is desirable to filter traffic to a particular group of devices or traffic that transits the network.

This type of filtering is traditionally performed by firewalls. However, there are instances where it may be beneficial to perform this filtering on a Cisco IOS device in the network, for example, where filtering must be performed but no firewall is present.

Transit ACLs are also an appropriate place in which to implement static anti-spoofing protections. See the Anti-Spoofing Protections section of this document for more information.

Refer to Transit Access Control Lists: Filtering at Your Edge for more information about tACLs.

ICMP Packet Filtering

The Internet Control Message Protocol (ICMP) was designed as a control protocol for IP. As such, the messages it conveys can have far reaching ramifications on the TCP and IP protocols in general. ICMP is used by the network troubleshooting tools **ping** and **traceroute**, as well as by Path MTU Discovery; however, external ICMP connectivity is rarely needed for the proper operation of a network.

Cisco IOS software provides functionality to specifically filter ICMP messages by name or type and code. This example ACL allows ICMP from trusted networks while it blocks all ICMP packets from other sources:

```
!

ip access-list extended ACL-TRANSIT-IN
!
!--- Permit ICMP packets from trusted networks only
!

 permit icmp host <trusted-networks> any
!
!--- Deny all other IP traffic to any network device
!

 deny icmp any any
!
```

Filter IP Fragments

As detailed previously in the Limit Access to the Network with Infrastructure ACLs section of this document, the filtering of fragmented IP packets can pose a challenge to security devices.

Because of the nonintuitive nature of fragment handling, IP fragments are often inadvertently permitted by ACLs. Fragmentation is also often used in attempts to evade detection by intrusion detection systems. It is for these reasons that IP fragments are often used in attacks and should be explicitly filtered at the top of any configured tACLs. The ACL below includes comprehensive filtering of IP fragments. The functionality illustrated in this example must be used in conjunction with the functionality of the previous examples:

```
!

ip access-list extended ACL-TRANSIT-IN
!
!--- Deny IP fragments using protocol-specific ACEs to aid in
!--- classification of attack traffic
!

 deny tcp any any fragments
 deny udp any any fragments
 deny icmp any any fragments
 deny ip any any fragments
!
```

Refer to Access Control Lists and IP Fragments for more information about ACL handling of fragmented IP packets.

ACL Support for Filtering IP Options

In Cisco IOS Software Release 12.3(4)T and later, Cisco IOS software supports the use of ACLs to filter IP packets based on the IP options that are contained in the packet. The presence of IP options within a packet might indicate an attempt to subvert security controls in the network or otherwise alter the transit characteristics of a packet. It is for these reasons that packets with IP options should be filtered at the edge of the network.

This example must be used with the content from previous examples to include complete filtering of IP packets that contain IP options:

```
!

ip access-list extended ACL-TRANSIT-IN
!
!--- Deny IP packets containing IP options
!

 deny ip any any option any-options
!
```

Anti-Spoofing Protections

Many attacks use source IP address spoofing to be effective or to conceal the true source of an attack and hinder accurate traceback. Cisco IOS software provides Unicast RPF and IP Source Guard (IPSG) in order to deter attacks that rely on source IP address spoofing. In addition, ACLs and null routing are often deployed as a manual means of spoofing prevention.

IP Source Guard works to minimize spoofing for networks that are under direct administrative control by performing switch port, MAC address, and source address verification. Unicast RPF provides source network verification and can reduce spoofed attacks from networks that are not under direct administrative control. Port Security can be used in order to validate MAC addresses at the access layer. Dynamic Address Resolution Protocol (ARP) Inspection (DAI) mitigates attack vectors that use ARP poisoning on local segments.

Unicast RPF

Unicast RPF enables a device to verify that the source address of a forwarded packet can be reached through the interface that received the packet. You must not rely on Unicast RPF as the only protection against spoofing. Spoofed packets could enter the network through a Unicast RPF-enabled interface if an appropriate return route to the source IP address exists. Unicast RPF relies on you to enable Cisco Express Forwarding on each device and is configured on a per-interface basis.

Unicast RPF can be configured in one of two modes: loose or strict. In cases where there is asymmetric routing, loose mode is preferred because strict mode is known to drop packets in these situations. During configuration of the **ip verify** interface configuration command, the keyword **any** configures loose mode while the keyword **rx** configures strict mode.

This example illustrates configuration of this feature:

```
!

ip cef
!

interface <interface>
  ip verify unicast source reachable-via <mode>
!
```

Refer to Understanding Unicast Reverse Path Forwarding for more information about the configuration and use of Unicast RPF.

IP Source Guard

IP Source Guard is an effective means of spoofing prevention that can be used if you have control over Layer 2 interfaces. IP Source Guard uses information from DHCP snooping to dynamically configure a port access control list (PACL) on the Layer 2 interface, denying any traffic from IP addresses that are not associated in the IP source binding table.

IP Source Guard can be applied to Layer 2 interfaces belonging to DHCP snooping-enabled VLANs. These commands enable DHCP snooping:

```
!

ip dhcp snooping
ip dhcp snooping vlan <vlan-range>
!
```

After DHCP snooping is enabled, these commands enable IPSG:

```
!
interface <interface-id>
   ip verify source
!
```

Port security can be enabled with the **ip verify source port security** interface configuration command. This requires the global configuration command **ip dhcp snooping information option**; additionally, the DHCP server must support DHCP option 82.

Refer to Configuring DHCP features and IP Source Guard for more information on this feature.

Port Security

Port Security is used in order to mitigate MAC address spoofing at the access interface. Port Security can use dynamically learned (sticky) MAC addresses to ease in the initial configuration. Once port security has determined a MAC violation, it can use one of four violation modes. These modes are protect, restrict, shutdown, and shutdown VLAN. In instances when a port only provides access for a single workstation with the use of standard protocols, a maximum number of one may be sufficient. Protocols that leverage virtual MAC addresses such as HSRP do not function when the maximum number is set to one.

```
!

interface <interface>
  switchport
  switchport mode access
  switchport port-security
  switchport port-security mac-address sticky
  switchport port-security maximum <number>
  switchport port-security violation <violation-mode>
!
```

Refer to Configuring Port Security for more information about the Port Security confuration.

Dynamic ARP Inspection

Dynamic ARP Inspection (DAI) can be used in order to mitigate ARP poisoning attacks on local segments. An ARP poisoning attack is a method in which an attacker sends falsified ARP information to a local segment. This information is designed in order to corrupt the ARP cache of other devices. Often an attacker uses ARP poisoning in order to perform a man-in-the-middle attack.

DAI intercepts and validates the IP-to-MAC address relationship of all ARP packets on untrusted ports. In DHCP environments, DAI uses the data that is generated by the DHCP snooping feature. ARP packets that are received on trusted interfaces are not validated and invalid packets on untrusted interfaces are discarded. In non-DHCP environments, the use of ARP ACLs is required.

These commands enable DHCP snooping:

```
!
ip dhcp snooping
ip dhcp snooping vlan <vlan-range>
!
```

Once DHCP snooping has been enabled, these commands enable DAI:

```
!
ip arp inspection vlan <vlan-range>
!
```

In non DHCP environments, ARP ACLs are required to enable DAI. This example demonstrates the basic configuration of DAI with ARP ACLs:

```
!

arp access-list <acl-name>
 permit ip host <sender-ip> mac host <sender-mac>
!

ip arp inspection filter <arp-acl-name> vlan <vlan-range>
!
```

Refer to Configuring Dynamic ARP Inspection for more information on how to configure DAI.

Anti-Spoofing ACLs

Manually configured ACLs can provide static anti-spoofing protection against attacks that use known unused and untrusted address space. Commonly, these anti-spoofing ACLs are applied to ingress traffic at network boundaries as a component of a larger ACL. Anti-spoofing ACLs require regular monitoring because they can frequently change. Spoofing can be minimized in traffic that originates from the local network if you apply outbound ACLs that limit the traffic to valid local addresses.

This example demonstrates how ACLs can be used in order to limit IP spoofing. This ACL is applied inbound on the desired interface. The ACEs that make up this ACL are not comprehensive. If you configure these types of ACLs, seek an up-to-date reference that is conclusive.

```
!

ip access-list extended ACL-ANTISPOOF-IN
 deny   ip 10.0.0.0 0.255.255.255 any
 deny   ip 192.168.0.0 0.0.255.255 any
!

interface <interface>
 ip access-group ACL-ANTISPOOF-IN in
!
```

Refer to Configuring Commonly Used IP ACLs for more information on how to configure Access Control Lists.

The official list of unallocated Internet addresses is maintained by Team Cymru. Additional information about filtering unused addresses is available at the Bogon Reference Page .

Limit CPU Impact of Data Plane Traffic

The primary purpose of routers and switches is to forward packets and frames through the device onward to final destinations. These packets, which transit the devices deployed throughout the network, can impact CPU operations of a device. The data plane, which consists of traffic that transits the network device, should be secured to ensure the operation of the management and control planes. If transit traffic can cause a device to process switch traffic, the control plane of a device can be affected which may lead to an operational disruption.

Features and Traffic Types that Impact the CPU

Although not exhaustive, this list includes types of data plane traffic that require special CPU processing and are process switched by the CPU:

- **ACL Logging** - ACL logging traffic consists of any packets that are generated due to a match (permit or deny) of an ACE on which the **log** keyword is used.

- **Unicast RPF** - Unicast RPF used in conjunction with an ACL might result in the process switching of certain packets.
- **IP Options** - Any IP packets with options included must be processed by the CPU.

- **Fragmentation** - Any IP packet that requires fragmentation must be passed to the CPU for processing.

- **Time-to-Live (TTL) Expiry** - Packets that have a TTL value less than or equal to 1 require Internet Control Message Protocol Time Exceeded (ICMP Type 11, Code 0) messages to be sent, which results in CPU processing.
- **ICMP Unreachables** - Packets that result in ICMP unreachable messages due to routing, MTU or filtering are processed by the CPU.
- **Traffic Requiring an ARP Request** - Destinations for which an ARP entry does not exist require processing by the CPU.

- **Non-IP Traffic** - All non-IP traffic is processed by the CPU.

See the General Data Plane Hardening section of this document for more information about Data Plane Hardening.

Filter on TTL Value

You can use the ACL Support for Filtering on TTL Value feature, introduced in Cisco IOS Software Release 12.4(2)T, in an extended IP access list to filter packets based on TTL value. This feature can be used in order to protect a device receiving transit traffic where the TTL value is a zero or one. Filtering packets based on TTL values can also be used in order to ensure that the TTL value is not lower than the diameter of the network, thus protecting the control plane of downstream infrastructure devices from TTL expiry attacks.

Note that some applications and tools such as **traceroute** use TTL expiry packets for testing and diagnostic purposes. Some protocols, such as IGMP, legitimately use a TTL value of one.

This ACL example creates a policy that filters IP packets where the TTL value is less than 6.

```
!
```

```
!--- Create ACL policy that filters IP packets with a TTL value
!--- less than 6
!

ip access-list extended ACL-TRANSIT-IN
 deny ip any any ttl lt 6
 permit ip any any
!
!--- Apply access-list to interface in the ingress direction
!

interface GigabitEthernet 0/0
 ip access-group ACL-TRANSIT-IN in
!
```

Refer to TTL Expiry Attack Identification and Mitigation for more information about filtering packets based on TTL value.

Refer to ACL Support for Filtering on TTL Value for more information about this feature.

In Cisco IOS Software Release 12.4(4)T and later, Flexible Packet Matching (FPM) allows an administrator to match on arbitrary bits of a packet. This FPM policy drops packets with a TTL value less than six.

```
!

load protocol flash:ip.phdf
!

class-map type access-control match-all FPM-TTL-LT-6-CLASS
 match field IP ttl lt 6
!

policy-map type access-control FPM-TTL-LT-6-DROP-POLICY
 class  FPM-TTL-LT-6-CLASS
  drop
!

interface FastEthernet0
 service-policy type access-control input FPM-TTL-LT-6-DROP-POLICY
!
```

Refer to Flexible Packet Matching, located on the Cisco IOS Flexible Packet Matching homepage, for more information about the feature.

Filter on the Presence of IP Options

In Cisco IOS Software Release 12.3(4)T and later, you can use the ACL Support for the Filtering IP Options feature in a named, extended IP access list in order to filter IP packets with IP options present. Filtering IP packets that are based on the presence of IP options can also be used in order to prevent the control plane of infrastructure devices from having to process these packets at the CPU level.

Note that the ACL Support for Filtering IP Options feature can be used only with named, extended ACLs. It should also be noted that RSVP, Multiprotocol Label Switching Traffic Engineering, IGMP Versions 2 and 3, and other protocols that use IP options packets might not be able to function properly if packets for these protocols are dropped. If these protocols are in use in the network, then the ACL Support for Filtering IP Options can be used; however, the ACL IP Options Selective Drop feature could drop this traffic and these protocols might not function properly. If there are no protocols in use that require IP options, ACL IP Options Selective Drop is the preferred method to drop these packets.

This ACL example creates a policy that filters IP packets that contain any IP options:

```
!

ip access-list extended ACL-TRANSIT-IN
 deny ip any any option any-options
 permit ip any any
!

interface GigabitEthernet 0/0
 ip access-group ACL-TRANSIT-IN in
!
```

This example ACL demonstrates a policy that filters IP packets with five specific IP options. Packets that contain these options are denied:

- 0 End of Options List (eool)
- 7 Record Route (record-route)
- 68 Time Stamp (timestamp)
- 131 - Loose Source Route (lsr)
- 137 - Strict Source Route (ssr)

```
!

ip access-list extended ACL-TRANSIT-IN
 deny ip any any option eool
 deny ip any any option record-route
 deny ip any any option timestamp
 deny ip any any option lsr
 deny ip any any option ssr
 permit ip any any
!

interface GigabitEthernet 0/0
 ip access-group ACL-TRANSIT-IN in
!
```

See the General Data Plane Hardening section of this document for more information about ACL IP Options Selective Drop.

Refer to Transit Access Control Lists: Filtering at Your Edge for more information about filtering transit and edge traffic.

Another feature in Cisco IOS software that can be used in order to filter packets with IP options is CoPP. In Cisco IOS Software Release 12.3(4)T and later, CoPP allows an administrator to filter the traffic flow of control plane packets. A device that supports CoPP and ACL Support for Filtering IP Options, introduced in Cisco IOS Software Release 12.3(4)T, may use an access list policy to filter packets that contain IP options.

This CoPP policy drops transit packets that are received by a device when any IP options are present:

```
!

ip access-list extended ACL-IP-OPTIONS-ANY
 permit ip any any option any-options
!

class-map ACL-IP-OPTIONS-CLASS
 match access-group name ACL-IP-OPTIONS-ANY
!

policy-map COPP-POLICY
 class ACL-IP-OPTIONS-CLASS
  drop
```

```
!

control-plane
 service-policy input COPP-POLICY
!
```

This CoPP policy drops transit packets received by a device when these IP options are present:

- 0 End of Options List (eool)
- 7 Record Route (record-route)
- 68 Time Stamp (timestamp)
- 131 Loose Source Route (lsr)
- 137 Strict Source Route (ssr)

```
!

ip access-list extended ACL-IP-OPTIONS
 permit ip any any option eool
 permit ip any any option record-route
 permit ip any any option timestamp
 permit ip any any option lsr
 permit ip any any option ssr
!

class-map ACL-IP-OPTIONS-CLASS
 match access-group name ACL-IP-OPTIONS
!

policy-map COPP-POLICY
 class ACL-IP-OPTIONS-CLASS
  drop
!

control-plane
 service-policy input COPP-POLICY
!
```

In the preceding CoPP policies, the access control list entries (ACEs) that match packets with the permit action result in these packets being discarded by the policy-map drop function, while packets that match the deny action (not shown) are not affected by the policy-map drop function.

Refer to Deploying Control Plane Policing for more information about the CoPP feature.

Control Plane Protection

In Cisco IOS Software Release 12.4(4)T and later, Control Plane Protection (CPPr) can be used in order to restrict or police control plane traffic by the CPU of a Cisco IOS device. While similar to CoPP, CPPr has the ability to restrict or police traffic using finer granularity than CoPP. CPPr divides the aggregate control plane into three separate control plane categories known as subinterfaces: Host, Transit, and CEF-Exception subinterfaces exist.

This CPPr policy drops transit packets received by a device where the TTL value is less than 6 and transit or non-transit packets received by a device where the TTL value is zero or one. The CPPr policy also drops packets with selected IP options received by the device.

```
!

ip access-list extended ACL-IP-TTL-0/1
 permit ip any any ttl eq 0 1
!
```

```
class-map ACL-IP-TTL-0/1-CLASS
 match access-group name ACL-IP-TTL-0/1
!

ip access-list extended ACL-IP-TTL-LOW
 permit ip any any ttl lt 6
!

class-map ACL-IP-TTL-LOW-CLASS
 match access-group name ACL-IP-TTL-LOW
!

ip access-list extended ACL-IP-OPTIONS
 permit ip any any option eool
 permit ip any any option record-route
 permit ip any any option timestamp
 permit ip any any option lsr
 permit ip any any option ssr
!

class-map ACL-IP-OPTIONS-CLASS
 match access-group name ACL-IP-OPTIONS
!

policy-map CPPR-CEF-EXCEPTION-POLICY
 class ACL-IP-TTL-0/1-CLASS
  drop
 class ACL-IP-OPTIONS-CLASS
  drop
!

!-- Apply CPPr CEF-Exception policy CPPR-CEF-EXCEPTION-POLICY to
!-- the CEF-Exception CPPr sub-interface of the device

!

control-plane cef-exception
 service-policy input CPPR-CEF-EXCEPTION-POLICY
!

policy-map CPPR-TRANSIT-POLICY
 class ACL-IP-TTL-LOW-CLASS
  drop
!

control-plane transit
 service-policy input CPPR-TRANSIT-POLICY
!
```

In the previous CPPr policy, the access control list entries that match packets with the permit action result in these packets being discarded by the policy-map drop function, while packets that match the deny action (not shown) are not affected by the policy-map drop function.

Refer to Understanding Control Plane Protection and Control Plane Protection for more information about the CPPr feature.

Traffic Identification and Traceback

At times, you can need to quickly identify and traceback network traffic, especially during incident response or poor network performance. NetFlow and Classification ACLs are the two primary methods to accomplish this with Cisco IOS software. NetFlow can provide visibility into all traffic on the network. Additionally, NetFlow can be implemented with collectors that can provide long-term trending and automated analysis.

Classification ACLs are a component of ACLs and require pre-planning to identify specific traffic and manual intervention during analysis. These sections provide a brief overview of each feature.

NetFlow

NetFlow identifies anomalous and security-related network activity by tracking network flows. NetFlow data can be viewed and analyzed via the CLI, or the data can be exported to a commercial or freeware NetFlow collector for aggregation and analysis. NetFlow collectors, through long-term trending, can provide network behavior and usage analysis. NetFlow functions by performing analysis on specific attributes within IP packets and creating flows. Version 5 is the most commonly used version of NetFlow, however, version 9 is more extensible. NetFlow flows can be created with sampled traffic data in high-volume environments.

CEF, or distributed CEF, is a prerequisite to enabling NetFlow. NetFlow can be configured on routers and switches.

This example illustrates the basic configuration of this feature. In previous releases of Cisco IOS software, the command to enable NetFlow on an interface is **ip route-cache flow** instead of **ip flow {ingress | egress}**.

```
!

ip flow-export destination <ip-address> <udp-port>
ip flow-export version <version>
!

interface <interface>
 ip flow <ingess|egress>
!
```

This is an example of NetFlow output from the CLI. The SrcIf attribute can aid in traceback.

```
router#show ip cache flow
IP packet size distribution (26662860 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .741 .124 .047 .006 .005 .005 .002 .008 .000 .000 .003 .000 .001 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .001 .007 .039 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 4456704 bytes
  55 active, 65481 inactive, 1014683 added
  41000680 ager polls, 0 flow alloc failures
  Active flows timeout in 2 minutes
  Inactive flows timeout in 60 seconds
IP Sub Flow Cache, 336520 bytes
  110 active, 16274 inactive, 2029366 added, 1014683 added to flow
  0 alloc failures, 0 force free
  1 chunk, 15 chunks added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet       11512      0.0        15    42      0.2      33.8      44.8
TCP-FTP           5606      0.0         3    45      0.0      59.5      47.1
TCP-FTPD          1075      0.0        13    52      0.0       1.2      61.1
TCP-WWW          77155      0.0        11   530      1.0      13.9      31.5
TCP-SMTP          8913      0.0         2    43      0.0      74.2      44.4
TCP-X              351      0.0         2    40      0.0       0.0      60.8
TCP-BGP            114      0.0         1    40      0.0       0.0      62.4
TCP-NNTP           120      0.0         1    42      0.0       0.7      61.4
TCP-other       556070      0.6         8   318      6.0       8.2      38.3
UDP-DNS         130909      0.1         2    55      0.3      24.0      53.1
UDP-NTP         116213      0.1         1    75      0.1       5.0      58.6
UDP-TFTP           169      0.0         3    51      0.0      15.3      64.2
```

```
UDP-Frag              1      0.0         1  1405      0.0       0.0      86.8
UDP-other         86247      0.1       226    29     24.0      31.4      54.3
ICMP              19989      0.0        37    33      0.9      26.0      53.9
IP-other            193      0.0         1    22      0.0       3.0      78.2
Total:          1014637      1.2        26    99     32.8      13.8      43.9

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP Pkts
Gi0/1         192.168.128.21  Local         192.168.128.20  11 CB2B 07AF   3
Gi0/1         192.168.150.60  Gi0/0         10.89.17.146    06 0016 101F  55
Gi0/0         10.89.17.146    Gi0/1         192.168.150.60  06 101F 0016   9
Gi0/1         192.168.150.60  Local         192.168.206.20  01 0000 0303  11
Gi0/0         10.89.17.146    Gi0/1         192.168.150.60  06 07F1 0016   1
```

Refer to Cisco IOS NetFlow for more information on NetFlow capabilities.

Refer to An Introduction to Cisco IOS NetFlow - A Technical Overview for a technical overview of NetFlow.

Classification ACLs

Classification ACLs provide visibility into traffic that traverses an interface. Classification ACLs do not alter the security policy of a network and are typically constructed to classify individual protocols, source addresses, or destinations. For example, an ACE that permits all traffic could be separated into specific protocols or ports. This more granular classification of traffic into specific ACEs can help provide an understanding of the network traffic because each traffic category has its own hit counter. An administrator might also separate the implicit deny at the end of an ACL into granular ACEs to help identify the types of denied traffic.

An administrator can expedite an incident response by using classification ACLs with the **show access-list** and **clear ip access-list counters** EXEC commands.

This example illustrates the configuration of a classification ACL to identify SMB traffic prior to a default deny:

```
!

ip access-list extended ACL-SMB-CLASSIFY
 remark Existing contents of ACL
 remark Classification of SMB specific TCP traffic
 deny   tcp any any eq 139
 deny   tcp any any eq 445
 deny   ip any any
!
```

In order to identify the traffic that uses a classification ACL, use the **show access-list** acl-name EXEC command. The ACL counters can be cleared by with the **clear ip access-list counters** acl-name EXEC command.

```
router#show access-list ACL-SMB-CLASSIFY
Extended IP access list ACL-SMB-CLASSIFY
    10 deny tcp any any eq 139 (10 matches)
    20 deny tcp any any eq 445 (9 matches)
    30 deny ip any any (184 matches)
```

Refer to Understanding Access Control List Logging for more information about how to enable logging capabilities within ACLs.

Access Control with VLAN Maps and Port Access Control Lists

VLAN Access Control Lists (VACLs), or VLAN maps and Port ACLs (PACLs), provide the capability to enforce access control on non-routed traffic that is closer to endpoint devices than access control lists that are applied to routed interfaces.

These sections provide an overview of the features, benefits, and potential usage scenarios of VACLs and PACLs.

Access Control with VLAN Maps

VACLs, or VLAN maps that apply to all packets that enter the VLAN, provide the capability to enforce access control on intra-VLAN traffic. This is not possible with ACLs on routed interfaces. For example, a VLAN map might be used in order to prevent hosts that are contained within the same VLAN from communication with each other, which reduces opportunities for local attackers or worms to exploit a host on the same network segment. In order to deny packets from using a VLAN map, you can create an access control list (ACL) that matches the traffic and, in the VLAN map, set the action to drop. Once a VLAN map is configured, all packets that enter the LAN are sequentially evaluated against the configured VLAN map. VLAN access maps support IPv4 and MAC access lists; however, they do not support logging or IPv6 ACLs.

This example uses an extended named access list that illustrates the configuration of this feature:

```
!

ip access-list extended <acl-name>
 permit <protocol> <source-address> <source-port> <destination-address>
     <destination-port>
!

vlan access-map <name> <number>
 match ip address <acl-name>
 action <drop|forward>
!
```

This example demonstrates the use of a VLAN map in order to deny TCP ports 139 and 445 as well as the vines-ip protocol:

```
!

ip access-list extended VACL-MATCH-ANY
 permit ip any any
!

ip access-list extended VACL-MATCH-PORTS
 permit tcp 192.168.1.0 0.0.0.255 192.168.1.0 0.0.0.255 eq 445
 permit tcp 192.168.1.0 0.0.0.255 192.168.1.0 0.0.0.255 eq 139
!

mac access-list extended VACL-MATCH-VINES
 permit any any vines-ip
!

vlan access-map VACL 10
 match ip address VACL-MATCH-VINES
 action drop
!

vlan access-map VACL 20
 match ip address VACL-MATCH-PORTS
 action drop
```

```
!

vlan access-map VACL 30
 match ip address VACL-MATCH-ANY
 action forward
!

vlan filter VACL vlan 100
!
```

Refer to Configuring Network Security with ACLs for more information about the configuration of VLAN maps.

Access Control with PACLs

PACLs can only be applied to the inbound direction on Layer 2 physical interfaces of a switch. Similar to VLAN maps, PACLs provide access control on non-routed or Layer 2 traffic. The syntax for PACLs creation, which takes precedence over VLAN maps and router ACLs, is the same as router ACLs. If an ACL is applied to a Layer 2 interface, then it is referred to as a PACL. Configuration involves the creation of an IPv4, IPv6, or MAC ACL and application of it to the Layer 2 interface.

This example uses an extended named access list in order to illustrate the configuration of this feature:

```
!

ip access-list extended <acl-name>
 permit <protocol> <source-address> <source-port> <destination-address>
     <destination-port>
!

interface <type> <slot/port>
 switchport mode access
 switchport access vlan <vlan_number>
 ip access-group <acl-name> in
!
```

Refer to the Port ACL section of Configuring Network Security with ACLs for more information about the configuration of PACLs.

Access Control with MAC

MAC access control lists or extended lists can be applied on IP network with the use of this command in interface configuration mode:

```
Cat6K-IOS(config-if)#mac packet-classify
```

Note: It is to classify Layer 3 packets as Layer 2 packets. The command is supported in Cisco IOS Software Release 12.2(18)SXD (for Sup 720) and Cisco IOS Software Releases 12.2(33)SRA or later.

This interface command has to be applied on the ingress interface and it instructs the forwarding engine to not inspect the IP header. The result is that you are able to use a MAC access list on the IP environment.

Private VLAN Use

Private VLANs (PVLANs) are a Layer 2 security feature that limits connectivity between workstations or servers within a VLAN. Without PVLANs, all devices on a Layer 2 VLAN can communicate freely. Networking situations exist where security can be aided by limiting communication between devices on a single VLAN. For example, PVLANs are often used in order to prohibit communication between servers in a publicly accessible subnet. Should a single server become compromised, the lack of connectivity to other servers due to the application of PVLANs might help limit the compromise to the one server.

There are three types of Private VLANs: isolated VLANs, community VLANs, and primary VLANs. The configuration of PVLANs makes use of primary and secondary VLANs. The primary VLAN contains all promiscuous ports, which are described later, and includes one or more secondary VLANs, which can be either isolated or community VLANs.

Isolated VLANs

The configuration of a secondary VLAN as an isolated VLAN completely prevents communication between devices in the secondary VLAN. There might only be one isolated VLAN per primary VLAN, and only promiscuous ports can communicate with ports in an isolated VLAN. Isolated VLANs should be used on untrusted networks like networks that support guests.

This configuration example configures VLAN 11 as an isolated VLAN and associates it to the primary VLAN, VLAN 20. The example below also configures interface FastEthernet 1/1 as an isolated port in VLAN 11:

```
!

vlan 11
 private-vlan isolated
!

vlan 20
 private-vlan primary
 private-vlan association 11
!

interface FastEthernet 1/1
 description *** Port in Isolated VLAN ***
 switchport mode private-vlan host
 switchport private-vlan host-association 20 11
!
```

Community VLANs

A secondary VLAN that is configured as a community VLAN allows communication among members of the VLAN as well as with any promiscuous ports in the primary VLAN. However, no communication is possible between any two community VLANs or from a community VLAN to an isolated VLAN. Community VLANs must be used in order to group servers that need connectivity with one another, but where connectivity to all other devices in the VLAN is not required. This scenario is common in a publicly accessible network or anywhere that servers provide content to untrusted clients.

This example configures a single community VLAN and configures switch port FastEthernet 1/2 as a member of that VLAN. The community VLAN, VLAN 12, is a secondary VLAN to primary VLAN 20.

```
!

vlan 12
 private-vlan community
!

vlan 20
 private-vlan primary
 private-vlan association 12
!
```

```
interface FastEthernet 1/2
 description *** Port in Community VLAN ***
 switchport mode private-vlan host
 switchport private-vlan host-association 20 12
!
```

Promiscuous Ports

Switch ports that are placed into the primary VLAN are known as promiscuous ports. Promiscuous ports can communicate with all other ports in the primary and secondary VLANs. Router or firewall interfaces are the most common devices found on these VLANs.

This configuration example combines the previous isolated and community VLAN examples and adds the configuration of interface FastEthernet 1/12 as a promiscuous port:

```
!

vlan 11
 private-vlan isolated
!

vlan 12
 private-vlan community
!

vlan 20
 private-vlan primary
 private-vlan association 11-12
!

interface FastEthernet 1/1
 description *** Port in Isolated VLAN ***
 switchport mode private-vlan host
 switchport private-vlan host-association 20 11
!

interface FastEthernet 1/2
 description *** Port in Community VLAN ***
 switchport mode private-vlan host
 switchport private-vlan host-association 20 12
!

interface FastEthernet 1/12
 description *** Promiscuous Port ***
 switchport mode private-vlan promiscuous
 switchport private-vlan mapping 20 add 11-12
!
```

When you implement PVLANs, it is important to ensure that the Layer 3 configuration in place supports the restrictions that are imposed by PVLANs and does not allow for the PVLAN configuration to be subverted. Layer 3 filtering with a Router ACL or firewall can prevent the subversion of the PVLAN configuration.

Refer to Private VLANs (PVLANs) - Promiscuous, Isolated, Community, located on the LAN Security homepage, for more information about the use and configuration of Private VLANs.

Conclusion

This document gives you a broad overview of the methods that can be used in order to secure a Cisco IOS system device. If you secure the devices, it increases the overall security of the networks that you manage. In this overview, protection of the management, control, and data planes is discussed, and recommendations for configuration are supplied. Where possible, sufficient detail is provided for the configuration of each associated feature. However, in all cases, comprehensive references are provided to supply you with the information needed for further evaluation.

Acknowledgments

Some feature descriptions in this document were written by Cisco information development teams.

Appendix: Cisco IOS Device Hardening Checklist

This checklist is a collection of all the hardening steps that are presented in this guide. Administrators can use it as a reminder of all the hardening features used and considered for a Cisco IOS device, even if a feature was not implemented because it did not apply. Administrators are advised to evaluate each option for its potential risk before they implement the option.

Management Plane

- Passwords
    - Enable MD5 hashing (secret option) for enable and local user passwords
    - Configure the password retry lockout
    - Disable password recovery (consider risk)
- Disable unused services
- Configure TCP keepalives for management sessions
- Set memory and CPU threshold notifications
- Configure
    - Memory and CPU threshold notifications
    - Reserve memory for console access
    - Memory leak detector
    - Buffer overflow detection
    - Enhanced crashinfo collection
- Use iACLs to restrict management access
- Filter (consider risk)
    - ICMP packets
    - IP fragments
    - IP options
    - TTL value in packets
- Control Plane Protection
    - Configure port filtering
    - Configure queue thresholds
- Management access
    - Use Management Plane Protection to restrict management interfaces
    - Set exec timeout
    - Use an encrypted transport protocol (such as SSH) for CLI access
    - Control transport for vty and tty lines (access class option)
    - Warn using banners
- AAA
    - Use AAA for authentication and fallback
    - Use AAA (TACACS+) for command authorization
    - Use AAA for accounting
    - Use redundant AAA servers
- SNMP
    - Configure SNMPv2 communities and apply ACLs
    - Configure SNMPv3

- Logging
    - Configure centralized logging
    - Set logging levels for all relevant components
    - Set logging source-interface
    - Configure logging timestamp granularity
- Configuration Management
    - Replace and rollback
    - Exclusive Configuration Change Access
    - Software resilience configuration
    - Configuration change notifications

Control Plane

- Disable (consider risk)
    - ICMP redirects
    - ICMP unreachables
    - Proxy ARP
- Configure NTP authentication if NTP is being used
- Configure Control Plane Policing/Protection (port filtering, queue thresholds)
- Secure routing protocols
    - BGP (TTL, MD5, maximum prefixes, prefix lists, system path ACLs)
    - IGP (MD5, passive interface, route filtering, resource consumption)
- Configure hardware rate limiters
- Secure First Hop Redundancy Protocols (GLBP, HSRP, VRRP)

Data Plane

- Configure IP Options Selective Drop
- Disable (consider risk)
    - IP source routing
    - IP Directed Broadcasts
    - ICMP redirects
- Limit IP Directed Broadcasts
- Configure tACLs (consider risk)
    - Filter ICMP
    - Filter IP fragments
    - Filter IP options
    - Filter TTL values
- Configure required anti-spoofing protections
    - ACLs
    - IP Source Guard
    - Dynamic ARP Inspection
    - Unicast RPF
    - Port security
- Control Plane Protection (control-plane cef-exception)
- Configure NetFlow and classification ACLs for traffic identification
- Configure required access control ACLs (VLAN maps, PACLs, MAC)
- Configure Private VLANs