Network Security Baseline

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-17300-01

Network Security Baseline

CONTENTS

CHAPTER 1

CHAPTER 2

CHAPTER 1

Introduction

Effective network security demands an integrated defense-in-depth approach. The first layer of a defense-in-depth approach is the enforcement of the fundamental elements of network security. These fundamental security elements form a security baseline, creating a strong foundation on which more advanced methods and techniques can subsequently be built.

Developing and deploying a security baseline can, however, be challenging due to the vast range of features available. The Network Security Baseline is designed to assist in this endeavour by outlining those key security elements that should be addressed in the first phase of implementing defense-in-depth. The main focus of Network Security Baseline is to secure the network infrastructure itself: the control and management planes.

This document outlines the key security elements identified for Network Security Baseline, along with implementation guidelines to assist in their design, integration, and deployment in production networks.

Security Baseline Overview

The Network Security Baseline presents the fundamental network security elements that are key to developing a strong network security baseline. The focus is primarily on securing the network infrastructure itself, as well as critical network services, and addresses the following key areas of baseline security:

- Infrastructure Device Access
- Routing Infrastructure
- Device Resiliency and Survivability
- Network Telemetry
- Network Policy Enforcement
- Switching Infrastructure

Unless these baseline security elements are addressed, additional security technologies and features are typically useless. For example, if a default access account and password are active on a network infrastructure device, it is not necessary to mount a sophisticated attack since attackers can simply log in to the device and perform whatever actions they choose.

In order to ensure a comprehensive solution, the Cisco Security Framework (CSF) is applied in the development of Network Security Baseline. CSF provides a comprehensive method of assessing and validating the security requirements of a system.

The CSF has been used in the creation of the Security Baseline to ensure that all the requirements have been considered for each particular contextual area. An overview of the CSF methodology is presented in the Cisco Security Framework Overview, page 1-2.

All sample configurations in this paper are based on Cisco IOS platforms and features. However, the general security objectives outlined in each section are equally applicable to non-IOS platforms.

Preliminary Network Design Assessment

The Network Security Baseline includes some security techniques that rely on the enforcement of IP address-based traffic filtering. These include ACLs to enforce policy on device management access, the ability to control route distribution and uRPF. More advanced security techniques, that can subsequently be added as an additional layer of security, also rely on IP address-based traffic filtering, such as firewall rule definition.

A rational, summarized, or compartmentalized IP address scheme, as well as the application of RFC1918 guidelines, makes the implementation of these IP address-based traffic filtering techniques simpler and more manageable on an ongoing basis.

In preparation for deploying a security baseline, it is recommended that a preliminary network design assessment be performed in order to facilitate its implementation. The key focus of this assessment is to review the current IP addressing scheme in terms of the following two key questions:

Q. Is the IP addressing scheme well structured and is it possible to easily summarize or compartmentalize the IP address space?

Q. Are RFC1918 IP addresses leveraged where appropriate?

An assessment of the current IP addressing scheme may identify areas where IP re-addressing may be desirable prior to implementation of a security baseline. Whilst this may demand some network changes, this will generally result in a more manageable and enforceable security policy, offering a significant benefit to overall network security.

For more information on RFC1918, see: http://www.ietf.org/rfc/rfc1918.txt

Cisco Security Framework Overview

The Cisco Security Framework (CSF) is a security operational process model aimed at ensuring network, and service, availability and business continuity. Security threats are an ever-moving target and the CSF is designed to identify current threat vectors, as well as track new and evolving threats, through the use of best common practices and comprehensive solutions.

The CSF is built upon two fundamental objectives, under the premise that one cannot control what one cannot see or measure:

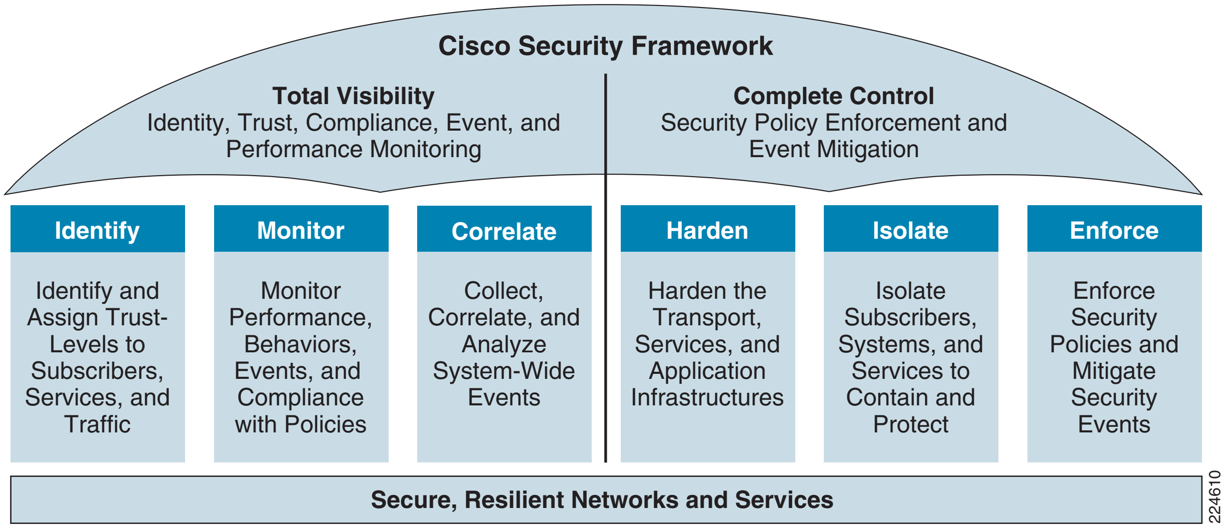
- Gain Total Visibility
- Identify, monitor, and correlate system-wide events
- Assure Complete Control

  Harden network infrastructure, isolate hosts and services, and enforce security policies

To achieve this total visibility and complete control, multiple technologies and capabilities are used throughout the network to gain visibility into network activity, enforce network policy, and address anomalous traffic. Network infrastructure elements such as routers and switches are leveraged as pervasive, proactive policy monitoring and enforcement agents.

The CSF focuses on six key actions, as illustrated in Figure 1-1.

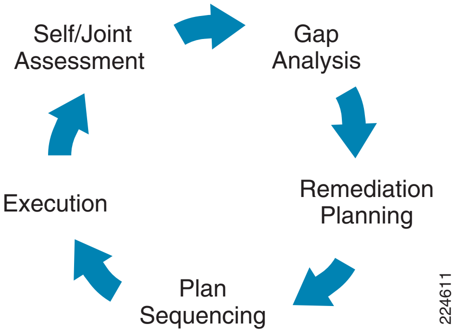
Figure 1-1 Cisco Security Framework Overview


Cisco Security Framework
Total Visibility
Identity, Trust, Compliance, Event, and Performance Monitoring
Complete Control
Security Policy Enforcement and Event Mitigation
Identify
Identify and Assign Trust-Levels to Subscribers, Services, and Traffic
Monitor
Monitor Performance, Behaviors, Events, and Compliance with Policies
Correlate
Collect, Correlate, and Analyze System-Wide Events
Harden
Harden the Transport, Services, and Application Infrastructures
Isolate
Isolate Subscribers, Systems, and Services to Contain and Protect
Enforce
Enforce Security Policies and Mitigate Security Events
Secure, Resilient Networks and Services
224610

The application of the CSF to a network results in the identification of technologies and best common practices to satisfy each one of the six key actions. However, the CSF is an ongoing process, involving review and modification of the implementation in accordance with changing business and security needs.

To that end, the CSF incorporates an evolutionary cycle, as illustrated in Figure 1-2.

Figure 1-2 CSF Evolution Cycle



The cycle starts with an initial assessment aimed at identifying current capabilities and security posture. It is followed by a gap analysis to unveil the strengths and weaknesses of the current architecture.

The Network Security Baseline presented may be used as a reference model during the initial assessment and gap analysis phases. It provides the minimum requirements for control and management protection. Strengths and weaknesses of real-world networks can be identified by comparing them against the baseline.

After the initial assessment and gap analysis, the cycle continues with remediation planning, which has the goal of closing the gap and satisfying future requirements by updating the overall network architecture. Plan sequencing follows to establish an implementation roadmap for the different components of the intended architecture. Each phase is then executed and results are evaluated as the cycle moves back into the assessment phase.

As Figure 1-2 illustrates, the process is iterative and each iteration results in the development of an architecture better designed to meet the evolving business and security policy needs.

The Network Security Baseline has been developed following the CSF. Each section includes a table showing how the proposed security features and best common practices help satisfy each one of the key actions of the CSF.

CHAPTER 2

Infrastructure Device Access

Securing the network infrastructure itself is critical to overall network security, be they routers, switches, servers, or other infrastructure devices. One key element of this is the security of management access to these infrastructure devices. If infrastructure device access is compromised, the security and management of the entire network can be compromised. Consequently, it is critical to establish the appropriate controls in order to prevent unauthorized access to infrastructure devices.

Network infrastructure devices often provide a range of different access mechanisms, including console and asynchronous connections, as well as remote access based on protocols such as Telnet, rlogin, HTTP, and SSH. Some mechanisms are typically enabled by default with minimal security associated with them; for example, Cisco IOS software-based platforms are shipped with console and modem access enabled by default. For this reason, each infrastructure device should be carefully reviewed and configured to ensure only supported access mechanisms are enabled and that they are properly secured.

The key steps to securing both interactive and management access to an infrastructure device are:

- **Restrict Device Accessibility**

  Limit the accessible ports, restrict the permitted communicators and restrict the permitted methods of access.

- **Present Legal Notification**

  Display legal notice, developed in conjunction with company legal counsel, for interactive sessions.

- **Authenticate Access**

  Ensure access is only granted to authenticated users, groups, and services.

- **Authorize Actions**

  Restrict the actions and views permitted by any particular user, group, or service.

- **Ensure the Confidentiality of Data**

  Protect locally stored sensitive data from viewing and copying. Consider the vulnerability of data in transit over a communication channel to sniffing, session hijacking and man-in-the-middle (MITM) attacks.

- **Log and Account for all Access**

  Record who accessed the device, what occurred, and when for auditing purposes.

Note It is critical to regularly review logs in order to audit access and identify any anomalous access attempts or actions.

CSF Methodology Assessment

The results of applying CSF to baseline infrastructure device access security are presented in Table 2-1 and Table 2-2. The tables highlight the technologies and features identified for baseline secure device management access and that are integrated in Network Security Baseline.

Total Visibility

Table 2-1 CSF Methodology Assessment—Total Visibility

Identify	Monitor	Correlate
• AAA Enforcement – Centralized AAA and local fallback – Administrator access – Privileged level access • SNMP Accounts – Community strings or auth/privacy policy • AAA server definitions • Device Management Best Common Practices – Strong password policy – Per-user accounts – Remove default accounts and passwords	• Logging – Syslog – SNMP – AAA Server Based Accounting – Configuration change notification and logging	

Complete Control

Table 2-2 CSF Methodology Assessment—Total Control

Harden	Isolate	Enforce
• SNMP • SSH/Telnet • HTTP/HTTPS • Restrict Device Accessibility – Transport types – VTY ACLs – SNMP ACLs – IOS login enhancements	• Console • Dedicated management interface • Management Network • SSH/Telnet • HTTP/HTTPS • ACLs • Out-of-band (OOB) Management	• Banners – MOTD – EULA • Local Password Protection – Password encryption – Secrets • File Transfer & Verification – FTP – TFTP – SCP – IOS image verification • Session Management • Device Management Best Common Practices – Minimum access privileges

Restrict Infrastructure Device Management Accessibility

The first step in securing management access to infrastructure devices is to restrict device accessibility. The key elements include:

- Restrict access to authorized terminal and management ports only
- Restrict access to authorized services and protocols only
- Restrict access attempts to authorized services by authorized originators only
- Only grant access to authenticated and authorized users
- Grant minimum privilege levels to authorized users
- Enforce session management
- Restrict vulnerability to dictionary and DoS attacks

The general approach to achieving each of these objectives is listed in Table 2-3. The general philosophy is that device management access should be implicitly denied and only permitted for those users and services that are explicitly required.

Table 2-3 Approaches for Infrastructure Device Management Accessibility Restriction

Infrastructure Device Management Accessibility Restriction	General Approach
Restrict access to authorized terminal and management ports only	• Disable all device terminal and management ports that are not explicitly required or actively being used for device management access
Restrict access to authorized services and protocols only	• Permit device management access only through required and supported services and protocols • Deny all other device management access services and protocols • Deny outgoing access unless explicitly required
Restrict access attempts to authorized services by authorized originators only	• Only permit access attempts to authorized services from authorized originators
Only grant access to authenticated and authorized users	• Use AAA to authenticate and authorize device management access on all supported ports and services
Enforce session management	• Enforce idle timeouts and keepalives to detect and close inactive or hung sessions. • Enforce an active session timeout to restrict the maximum duration of a session prior to re-authentication
Restrict vulnerability to dictionary and DoS attacks	• Limit the rate of login attempts • Enforce a lockout period upon multiple authentication failure attempts • Reserve one management port for access only by one particular NoC host

Note Most infrastructure devices can be accessed through a variety of terminal and management ports, services and protocols, some of which may be enabled by default. All possible management access mechanisms should be reviewed and secured.

Cisco IOS Device Interactive Terminal and Management Access Lines

Cisco IOS software-based platforms typically offer interactive device management access through the following ports and lines:

- TTY lines

  Asynchronous ports, including:
  - AUX
  - console
- VTY lines

Virtual TTY lines used for remote access such as:

- Telnet
- SSH
- rlogin

Note that web-based GUI (HHTP/HTTPS) and SNMP access are covered in subsequent sections.

AUX Port

Interactive access via an AUX port is typically used to provide either dial-in or dial-out management access to a platform. If this is not required, the line should be disabled to reduce the risk of unauthorized access.

Console Port

Interactive access via a console port is directly accessible by a local user or remotely accessible through the use of a terminal or console server. If console port access is required, the line should be properly secured to prevent unauthorized access.

Note If a terminal server is employed, it is vital to ensure that this device is properly secured by enforcing the security guidelines presented in this paper.

VTY Line

Interactive access via a VTY line is the most commonly used method to remotely manage a device. If VTY access is required, the lines should be properly secured to prevent unauthorized access.

Sample TTY and VTY configurations are provided in Appendix A, "Sample Configurations."

Note A router typically has 5 VTY lines (VTY 0 4) but more may be supported. It is critical to ensure that security guidelines are applied to all available VTY lines.

Disable Unnecessary Device Terminal and Management Access Ports

Some network infrastructure devices have terminal and management ports and interfaces enabled by default. This can present a security risk. It is recommended to disable all terminal and management ports and interfaces which are not required or are not used.

On a Cisco IOS device, terminal and management ports typically include TTY and VTY lines. These ports can be disabled using the **no exec** command as shown in the following configuration:

```
!
! Disable access to VTY
line vty 1
 login
 no exec
!
! Disable access to Console
line con 0
  no exec
```

```
!
```

Restrict Device Access to Authorized Services and Protocols Only

Some network infrastructure devices have device management access services and protocols enabled by default. This can present a security risk. It is recommended to disable all device management access services and protocols that are not required or are not used.

On a Cisco IOS device, device management access services and protocols typically include:

- Interactive access via Telnet, SSH, etc.
- HTTP, HTTPS
- SNMP

Interactive access, through the TTY and VTY lines of a Cisco IOS device, should be restricted to only those authorized access services and protocols required and permitted, per corporate security policy. Restrictions should be enforced on both incoming and outgoing connections.

This is enforced on TTY and VTY lines using the **transport** command. Some examples are provided in Table 2-4.

Table 2-4 Examples of Restricting Incoming and Outgoing Connections

Cisco IOS TTY and VTY Line Device Management Access Protocol Restriction	IOS Configuration on TTY or VTY
No incoming connections	`transport input none`
No outgoing connections	`transport output none`
Only SSH permitted for incoming connections	`transport input ssh`
Only telnet permitted for incoming connections	`transport input telnet`
SSH or telnet permitted for incoming connections	`transport input telnet ssh`
Only SSH permitted for outgoing connections	`transport output ssh`
Transport protocol must be specified in access request	`transport preferred none`

Note The best practice is to prefer encrypted access protocols, such as SSH, over clear text protocols like Telnet.

SSH is covered in more detail in Secure Shell (SSH), page 2-18.

Security guidelines for HTTP, HTTPS, and SNMP are described at following:

- HTTP, page 2-20
- HTTPS, page 2-21
- SNMP Access, page 2-21

For more information on the **transport** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/termserv/command/reference/ter_t1g.html#wp1083564

Restrict Device Access Attempts To Authorized Services By Authorized Originators Only

Only authorized originators should be permitted to even attempt device management access, and only to the services they are authorized to use. This ensures that the processing of access requests is restricted to only authorized services by an authorized source IP address.This reduces the risk of unauthorized access and the exposure to other attacks, such as brute force, dictionary, or DoS attacks.

On a Cisco IOS device, standard ACLs can be used to restrict device management access attempts to authorized originators only. Extended ACLs can be used to restrict device management access attempts to authorized services by authorized originators only.

It should be noted that the more restrictive an ACL, the more limited the exposure to unauthorized access attempts. However, a more restrictive ACL, such as one restricting access to NoC hosts only, can create a management overhead, can impact accessibility if network connectivity is lost for a specified location and can limit the ability of authorized administrators to perform troubleshooting, such as those at a remote location investigating local anomalies. Consequently, there is a balance to be considered. One compromise is to restrict access to internal corporate IP addresses only.

Note Each customer must evaluate the implementation of ACLs in relation to their own security policy, risks, exposure, and acceptance thereof.

The use of ACLs to restrict traffic directed to an infrastructure device itself is covered in more detail in the Appendix 4, "Device Resiliency and Survivability."

Standard ACLs

Standard ACLs allow restrictions to be enforced on the originator source IP address or IP address range.

```
!
access-list 10 permit <NOCsubnet> <inverse-mask>
access-list 10 deny any any
!
```

Extended ACLs

Extended ACLs allow restrictions to be enforced on the originator source IP address or IP address range, and the access protocol.

```
!
access-list <xACL#> permit tcp <NOCsubnet1> <inverse-mask> any eq <TCP port>
access-list <xACL#> permit tcp <NOCsubnet2> <inverse-mask> any eq <TCP port>
access-list <xACL#> deny ip any any log-input
!
```

Note Access-class ACLs only support the any clause as destination.

The ACL must subsequently be configured on the appropriate lines, services, and interfaces in order for it to be enforced. An example of how to enforce an ACL on VTY lines is shown below:

```
!
line vty 0 4
 access-class <ACL#> in
!
```

Note A highly restrictive ACL can also be applied to one VTY in order to try to preserve interactive access during a DoS attack on VTY lines. For more information, refer to Restrict Login Vulnerability to Dictionary and DoS Attacks, page 2-10.

ACL enforcement for SNMP and HTTP access is addressed in their related sections (see SNMP Access, page 2-21 and HTTP, page 2-20).

Enforce Device Login Authentication Using AAA

Access to all infrastructure device management ports should be authenticated to restrict access to authorized users only. It is recommended that a centralized AAA server be deployed to enforce per-user, AAA-based login authentication on all infrastructure device terminal and management ports.

In Cisco IOS, administrative access to a network infrastructure device is referred to as an EXEC session and is performed over a TTY or VTY line. AAA-based authentication of EXEC user login is enforced by applying a AAA method list to all available TTY and VTY lines.

An example IOS configuration for the enforcement of AAA-based authentication, with local fallback, for EXEC user login on the console and VTY lines is shown below:

```
!
aaa authentication login adminAuthen-list group adminAAAgroup local-case
!
line con 0
 login authentication adminAuthen-list
!
line vty 0 4
 login authentication adminAuthen-list
!
```

For more information on Cisco IOS named method lists, refer to AAA Method Lists, page 2-16.

For more information on the AAA authentication login command, refer to the following URL:
http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfathen.html#wp1001032

Enforce Device Login Authorization Using AAA

Only minimum access privileges should be granted to authenticated users, according to their specific access requirements. This reduces exposure to both malicious and unintentional security incidents.

In Cisco IOS, the ability to control which users are authorized to open an EXEC session and access the CLI is achieved using the **aaa authorization exec** command. In conjunction with AAA, different user groups can easily be granted different access privileges.

An example of AAA-based EXEC session access authorization on VTY lines is shown below, including fallback to authorization being granted if a user is authenticated, in case a AAA server is not available.

```
!
aaa authorization exec adminAuthor-list group adminAAAgroup if-authenticated
line vty 0 4
 authorization exec adminAuthor-list
!
```

Note The AAA server must set the 'Service-Type' attribute to EXEC (login) in order to grant EXEC session access.

For more information on the **aaa authorization exec** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/security/command/reference/sec_a1g.html#wp1071720

Enforce Privileged Level Authentication Using AAA

It is critical to ensure that an administrative user attempting to obtain privileged level access is properly authenticated. Privileged level access typically refers to a level of access which provides the ability to configure a network infrastructure device.

In Cisco IOS, privileged level EXEC access may be obtained either from the CLI using the **enable** command, or automatically as a result of RADIUS or TACACS+ authorization. The later requires exec authorization and the configuration of the privilege level at the user or group profile on the AAA server.

Privilege level access extends the access level of an EXEC session, providing the ability to configure the device. Therefore, in accordance with the enforcement of minimum access privileges, enable access should only be granted to those users requiring this level of access.

Cisco IOS enable access should be authenticated using AAA-based authentication to a centralized AAA server with local fallback to the enable secret. This is achieved by defining a default AAA method list for enable authentication.

```
!
aaa authentication enable default group adminAAAgroup enable
!
```

It is recommended that an **enable secret** be configured instead of an **enable password**, since the enable secret provides Type 5 encryption which is not reversible. See Enable Secret, page 2-24, for more details.

For more information on the **aaa authentication enable default** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/11_3/security/configuration/guide/scauthen.html#wp6292

Enforce Session Management

Device access sessions should be managed to ensure that the following scenarios are addressed:

- Idle sessions
- Hung sessions

Idle Sessions

Idle sessions should not be permitted to consume a terminal or management port indefinitely. This preserves the availability of terminal and management ports, and reduces the exposure to session hijacking.

In Cisco IOS, an idle timeout is configurable on TTY and VTY lines with the command session-timeout. By default, a VTY session has a 10 minute idle timeout.

```
Router(config-line)# session-timeout <minutes>
```

The session-timeout command behaves slightly differently on virtual (VTY) terminals than on physical console, auxiliary (AUX), and terminal (TTY) lines. When a timeout occurs on a VTY, the user session returns to the EXEC prompt. When a timeout occurs on physical lines, the user session is logged out and the line returned to the idle state.

You can use a combination of the exec-timeout and session-timeout line configuration commands, set to approximately the same values, to get the same behavior from virtual lines that the session-timeout command causes on physical lines.

In Cisco IOS, by default a VTY session has a 10 minute exec timeout.

```
Router(config-line)# exec-timeout <minutes> [seconds]
```

For more information on the **session-timeout** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/termserv/command/reference/ter_l1g.html#wp1037637

For more information on the **exec-timeout** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g03.html#wp1029531

Hung Sessions

If a remote system crashes while a management session is in progress, the session may remain open and vulnerable to attack. Consequently, hung sessions should be detected and closed in order to preserve the availability of terminal and management ports and reduce the exposure to session hijacking.

In Cisco IOS, hung sessions on VTY lines can be detected and closed with the **service tcp-keepalives-in** command. This causes TCP keepalives to be sent on incoming connections, enabling a remote system crash to be detected if no response is received.

```
Router(config)# service tcp-keepalives-in
```

For more information on the **service tcp-keepalives-in** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g07.html#wp1029289

Restrict Login Vulnerability to Dictionary and DoS Attacks

The vulnerability of a network infrastructure device management access to dictionary and DoS attacks may be reduced by enforcing the following restrictions:

- Enforce the use of strong passwords
- Restrict the frequency of login attempts
- Restrict the number of login failures permitted within a specified time period
- Reserve a terminal or management port

The features available to enforce these requirements are listed Table 2-5.

Table 2-5 Features for Restricting Login Vulnerability to Dictionary and DoS Attacks

Requirement	Implementation Options
Enforce the use of strong passwords	AAA server feature: Enforce the use of strong passwords on the AAA server, in compliance with the security policy
	Cisco IOS feature to force a minimum password length[1]: `security passwords min-length`
Restrict the frequency of login attempts	Cisco IOS login enhancement feature[2]: `login delay`
Restrict the number of login failures permitted within a specified time period	AAA server feature: Enforce account lockout on the AAA server if a defined number of failed login attempts within a specified time period is exceeded

	Cisco IOS login enhancements [2, 3]
	`login block-for`
	`login quiet-mode access-class`

Reserve one terminal or management port	Cisco IOS feature: Highly restrictive ACL on last VTY line
Log and monitor user login authentication failures	See Infrastructure Device Management Access Logging, page 2-25.

[1] The Cisco IOS feature to force a minimum password length was introduced in 12.3(1) and was integrated into Cisco IOS software Release 12.2(18)T.

[2] The Cisco IOS login enhancement feature was introduced in 12.3(4)T and 12.2(25)S.

[3] The Cisco IOS login enhancement feature to restrict the number of login failures permitted within a specified time period is typically only used when a AAA server is not being employed to enforce authentication.

For more information about Cisco IOS login enhancements, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_login_enhance_ps6350_TSD_Products_Configuration_Guide_Chapter.html

Enforce The Use of Strong Passwords

In the event of a dictionary attack, the use of strong passwords makes such an attack less likely to succeed, since the passwords will not be simple dictionary words. If a AAA server is employed for login authentication, the AAA server typically offers a feature to enforce the use of strong passwords, according to the security policy. If a AAA server is not available to enforce a strong password policy, local device features to minimize vulnerability to dictionary attacks should be employed, as available.

If features to address password vulnerability to dictionary attacks are not available, a basic feature that may be available is the enforcement of a minimum password length. Whilst this type of feature does not provide direct protection against dictionary attacks, it provides protection against the use of commonly guessed passwords such as **cisco** and **lab**.

Cisco IOS Minimum Password Length Feature

Cisco IOS offers the ability to enforce a minimum password length for user passwords, enable passwords, enable secrets, and line passwords. This feature is enabled with the global configuration command:

```
Router(config)# security passwords min-length length
```

Once this command is enabled, any password that is less than the specified number of characters will fail.

Note This feature does not provide any protection against dictionary attacks.

For more information about the **security passwords min-length** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/security/command/reference/sec_r1g.html#wp1081544

Restrict Frequency of Login Attempts

In the event of a dictionary attack, introducing a delay between login attempts slows down the attack, increasing the time required for the attack to succeed and the timeframe available for the anomaly to be identified and addressed.

In Cisco IOS, the introduction of a delay between successive login attempts can be achieved using the global configuration **login delay** command. The default is a 1 second delay.

```
Router(config)# login delay <seconds>
```

For more information on the **login-delay** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_k1.html#wp1031384

Restrict Number of Login Failures Permitted Within Specified Time Period

In the event of a dictionary attack, restricting the maximum number of failed access attempts within a specified period can slow down the attack, increasing the time required to succeed and the timeframe available for the anomaly to be identified and addressed.

If a AAA server is employed for login authentication, the AAA server typically offers a feature, enforcing account lockout if a defined number of failed login attempts occur within a specified time period. If a AAA server is not employed, then the Cisco IOS feature may be employed.

In Cisco IOS, the definition of the maximum number of failed login attempts permitted within a specified time period, after which the IOS device will not accept any additional connection attempts for a configurable "quiet period", can be achieved using the global configuration **login block-for** command as follows:

```
Router(config)# login block-for seconds attempts tries within seconds
```

For more information on the **login block-for** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_k1.html#wp1031219

The Cisco IOS also offers the ability to define an exception ACL for trusted systems and networks from which legitimate connections are expected. This exception ACL can be defined with the **login quiet-mode access-class** global command:

```
Router (config)# login quiet-mode access-class
```

For more information on the **login quiet-mode access-class** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_k1.html#wp1031806

The following example shows how to configure a router to enter a 100-second quiet period if 15 failed login attempts is exceeded within 100 seconds; all login requests will be denied during the quiet period except from the host defined in ACL 10.

```
Router(config)# access-list 10 permit host 172.26.150.206
Router(config)# login block-for 100 attempts 15 within 100
Router(config)# login quiet-mode access-class 10
```

Reserve a Terminal or Management Port

A DoS attack on an infrastructure device can target the terminal and management ports. This type of attack relies on the fact that there are only a limited number of terminal and management ports available and that, once all ports are in use, even if the connection has not yet been authenticated, no additional connections can be established.

Cisco IOS software devices have only a limited number of VTY lines, typically five. When all VTY lines are in use, no more remote interactive connections can be established. This creates an opportunity for a DoS attack if an attacker can open remote sessions to all VTYs available on a system, preventing an authorized administrator from gaining access. The attacker does not need to log in to achieve this type of DoS attack, the remote sessions can simply be left at the login prompt. The use of AAA does not mitigate this type of attack as the attacker does not need to attempt a login, it is only necessary to maintain a connection to the port, thus rendering it unavailable to other users.

One way to address this type of attack is to enforce highly restrictive access controls on one terminal or management port to preserve availability during this type of DoS attack. For example, this port can only be accessed by one particular NoC host.

In Cisco IOS, this may be achieved by configuring a highly restrictive ACL on the last VTY. The last VTY, usually VTY 4, can be restricted to accept connections only from a single, specific administrative station, such as a highly secured NoC host, whereas the other VTYs accept connections from any address in a wider address range, such as the NoC.

```
!
access-list 10 permit <NOCsubnet> <inverse-mask>
access-list 20 permit host <NOC-Host>
line vty 0 3
     access-class 10 in
line vty 4
     access-class 20 in
!
```

Legal Notification Banners

It is recommended that a legal notification banner is presented on all interactive sessions to ensure that users are notified of the security policy being enforced and to which they are subject.

In some jurisdictions, civil and/or criminal prosecution of an attacker who breaks into a system is easier, or even required, if a legal notification banner is presented, informing unauthorized users that their use is in fact unauthorized. In some jurisdictions, it may also be forbidden to monitor the activity of an unauthorized user unless they have been notified of the intent to do so.

Legal notification requirements are complex and vary in each jurisdiction and situation. Even within jurisdictions, legal opinions vary, and this issue should be discussed with your own legal counsel to ensure that it meets company, local and international legal requirements. This is often critical to securing appropriate action in the event of a security breach.

In cooperation with the company legal counsel, statements which may be included in a legal notification banner include:

- Notification that system access and use is permitted only by specifically authorized personnel, and perhaps information about who may authorize use.
- Notification that unauthorized access and use of the system is unlawful, and may be subject to civil and/or criminal penalties.
- Notification that access and use of the system may be logged or monitored without further notice, and the resulting logs may be used as evidence in court.
- Additional specific notices required by specific local laws.

From a security, rather than a legal, point of view, a legal notification banner should not contain any specific information about the device, such as its name, model, software, location, operator or owner because this kind of information may be useful to an attacker.

A sample legal notification banner is provided in Appendix A, "Sample Configurations."

In Cisco IOS, a number of banner options are available, including **banner motd**, **banner login**, **banner incoming**, and **banner exe**c.

When a user connects to an IOS device, a message-of-the-day (MOTD) banner, if configured, will appear first, followed by a login banner (if configured) and a login prompt. After a user successfully logs in to an IOS device, an incoming banner will be displayed for a reverse Telnet login and an EXEC banner will be displayed for all other types of connections.

It is recommended that either a MOTD or a login banner is implemented to ensure that a legal notification banner is presented on all device management access sessions, prior to a login prompt being presented.

For more information about the **banner login** or the **banner motd** commands, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g01.html#wp1029811

AAA Services

AAA Overview

AAA is an architectural framework for configuring a set of three independent security functions in a consistent, modular manner:

- **Authentication**

  Enables a user to be identified and verified prior to them being granted access to the network and network services.
- **Authorization**

  Defines the access privileges and restrictions to be enforced for an authenticated user.
- **Accounting**

Provides the ability to track user access, including user identities, start and stop times, executed commands (such as command line interface (CLI) commands), number of packets, and number of bytes.

AAA is the primary and recommended method for access control. Cisco IOS software provides additional features for simple access control, such as local username authentication and line password authentication, however, these features do not provide the same degree of access control that is possible with AAA and are not recommended, even for small deployments. Even if a separate AAA server is not being deployed, AAA to the local database should used on the Cisco IOS device. See the section on Centralized AAA for more information on the value of AAA to a local database.

AAA authentication, authorization and accounting are enforced by applying named method lists to access interfaces. Method lists are covered in detail in a subsequent section.

AAA uses protocols such as RADIUS, TACACS+, or Kerberos to administer its security functions.

For more information on AAA services, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_aaa_overview_ps6350_TSD_Products_Configuration_Guide_Chapter.html

Centralized AAA

The recommended method of administering AAA is on a centralized AAA server with local passwords as a fallback method. Local fallback provides a method of authentication in case communication with the AAA server is not possible. The key benefits of using a centralized AAA server include:

- **Manageability**

  Usernames and passwords are stored in a separate, central location which may be independently managed and leveraged across multiple devices.

- **Scalability**

  The AAA server(s) may be independently scaled according to the size of the user database and the number of transactions per second.

- **Security**

  Company-wide usernames and passwords may be stored off the router in a secure, encrypted file system or database. In contrast, locally stored passwords on Cisco IOS devices, even if encrypted, are still reversible.

- **Accountability**

  Access attempts and authorized sessions may be independently logged on the AAA server

If a centralized AAA server is not currently required or deployed, it is still recommended to implement authentication using a AAA configuration, even though a local user password database will be used. This enables the implementation of per-user local passwords, rather than all users using the same login secret or password. This approach offers greater security, visibility and control, along with easier migration to a possible future deployment leveraging a centralized AAA server.

AAA Server Groups

In Cisco IOS, a AAA server group is a list of AAA server hosts of a particular type, e.g. RADIUS or TACACS+, which are used to perform AAA. The particular AAA server group to be used for each particular AAA service is defined by the AAA method list, as discussed below.

The use of the AAA server-group feature provides greater flexibility and control over which AAA servers are used for which purposes, as well as offering redundancy across the defined servers.

For example, different AAA servers may be used for different AAA services to enable the separation and prioritization of device access management from end-user access management through the use of two independently maintained and scaled data stores. For example, infrastructure device access management may be authenticated using a set of TACACS+ servers, whereas end-user network access may be authenticated using a set of RADIUS servers.

```
!
aaa group server tacacs+ adminAAAgroup
 server TAC+server1
 server TAC+server2
!
aaa group server radius enduserAAAgroup
 server RADserver1
 server RADserver2
!
```

For more information on the **aaa group server** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_a1.html#wp1045019

AAA Method Lists

In Cisco IOS, AAA is enforced through the definition of named method lists which are applied to access interfaces. A method list is a sequential list that defines the authentication or authorization methods to be enforced and the sequence in which they will be attempted. Method lists enable one or more security protocols to be used for authentication or authorization, ensuring availability of a backup system in case an initial method is not available.

Cisco IOS software will first attempt the first method listed; if that method does not respond, the next method in the method list will be attempted. This process continues until there is successful communication with a listed method or the method list is exhausted, in which case authentication or authorization fails.

A sample named authentication method list with the name **admin-list** , whose first method is to attempt authentication to the TACACAS+ servers in the server group **adminAAAgroup**, falling back to local authentication if those servers are not available is shown below:

```
aaa authentication login adminAuthen-list group adminAAAgroup local-case
aaa group server tacacs+ adminAAAgroup
 server TAC+server1
 server TAC+server2
```

Note Cisco IOS software attempts authentication or authorization with the next listed method only when there is no response from the previous method. If authentication or authorization fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the process stops and no other methods are attempted.

A AAA method list must be applied to an access line before it will be enforced. The only exception to this is a default AAA method list (which is named **default**). A named method list is automatically applied to all access lines if no other method list is applied. A defined method list overrides the default method list.

```
aaa authentication login default group enduserAAAgroup local-case
aaa group server radius enduserAAAgroup
 server RADserver1
 server RADserver2
```

For more information on AAA method lists and groups, refer to the following URLs:

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_aaa_overview_ps6350_TSD_Products_Configuration_Guide_Chapter.html#wp1000933

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_cfg_radius_ps6350_TSD_Products_Configuration_Guide_Chapter.html#wp1001168

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_cfg_tacacs+_ps6350_TSD_Products_Configuration_Guide_Chapter.html#wp1001011

AAA Server Communication Security

Communication between an authenticator (also referred to as a NAS, Network Access Server) and a AAA server is commonly performed using RADIUS or TACACS+. The security of this communication can be summarized as follows:

- Both RADIUS and TACACS+ transactions are authenticated using a shared, static secret (or key) associated with the device name or IP address but this secret is never sent over the network.
- RADIUS, per the standard, only encrypts the user password field. All other packet data is passed in clear text and is thus vulnerable to sniffing.
- TACACS+ encrypts the full payload of the packet, thereby providing some confidentiality of data, though the encryption algorithm is not very strong.

The general guidelines for securing AAA server communication are:

- Employ strong secrets for authentication of the AAA server and NAS.
- Regularly change the secrets used to authenticate the AAA server and NAS.
- Restrict AAA communication to the limited set of authorized AAA servers, and over the configured AAA communication ports, using extended ACLs.

  The use of ACLs to restrict traffic directed to an infrastructure device itself is covered in more detail in Appendix 4, "Device Resiliency and Survivability."
- Since RADIUS and TACACS+ do not support strong authentication and encryption, it is recommended that an out-of-band (OOB) or IPSec management network be considered as a means of protecting AAA server communication transactions from attack.

AAA Server Based Accounting Services

It is critical to ensure that device management access is logged. This is covered in more detail in General Device Access and Configuration Change Logging Best Common Practices, page 2-30, but one method of logging device management access is using AAA server based accounting.

AAA server-based accounting enables the ability to track the services users are accessing, as well as the amount of network resources they are consuming. When AAA server-based accounting is enabled, the network infrastructure device reports user activity to the AAA server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the AAA server. This data can then be analyzed for network management, client billing, and/or auditing.

Cisco IOS software supports five different kinds of accounting:

- Network Accounting

  Network accounting provides information for all PPP, SLIP, or ARAP sessions, including packet and byte counts.

- **Connection Accounting**

  Connection accounting provides information about all outbound connections made from the network access server, such as Telnet, local-area transport (LAT), TN3270, packet assembly-disassembly (PAD), and rlogin.

- **EXEC Accounting**

  EXEC accounting provides information about user EXEC terminal sessions (user shells) on the network access server, including username, date, start and stop times, the access server IP address, and (for dial-in users) the telephone number the call originated from.

- **Command Accounting**

  Command accounting provides information about the EXEC shell commands for a specified privilege level that are being executed on a network access server. Each command accounting record includes a list of the commands executed for that privilege level, as well as the date and time each command was executed, and the user who executed it.

- **System Accounting**

  System accounting provides information about all system-level events (for example, when the system reboots or when accounting is turned on or off).

The Network Security Baseline is focused on securing the network infrastructure and critical network services. Consequently, AAA-based accounting in Network Security Baseline includes:

- EXEC accounting
- Command accounting
- System accounting

These elements are covered in detail in Infrastructure Device Management Access Logging, page 2-25.

Secure Shell (SSH)

SSH is a protocol that provides secure remote access, remote command execution, and file transfer. SSH implements strong authentication and encryption, making it a better option over insecure protocols such as rlogin and Telnet.

There are two versions of SSH: v1 and v2. SSHv2 addresses a series of security issues found in the v1. For these reasons, v2 should be used whenever it is supported. Cisco IOS supports both versions of SSH.

SSH authentication supports a variety of protocols including TACACS+, RADIUS, and RSA authentication. SSH also provides support for a wide range of encryption ciphers such as DES, 3DES, IDEA, RC4-128, and others. In addition, SSH can tunnel TCP connections, which allows not only securing login sessions, but also email and file transfers with secure copy (SCP) and secure FTP (SFTP).

The following steps are required to enable SSH support on an IOS device:

Step 1 Configure a hostname and DNS domain for the router.

Step 2 Generate an RSA key pair.

Step 3 Optionally, configure time-out and number of authentication retries. By default, the authentication timeout is set to 120 seconds and authentication retries to three attempts.

Step 4 Limit VTYs to SSH only (Highly recommended).

Step 5 Restrict SSH access to trusted hosts or subnets.

Note On some Cisco IOS platforms, SSH requires an IPSec (DES or 3DES) encryption IOS software image.

The following example shows how SSH can be configured on a Cisco IOS device:

```
!--- Step 1: Configure a hostname and domain name
Router(config)# hostname router
Router (config)# ip domain-name nyc.cisco.com

!--- Step 2: Generate an RSA key pair, automatically enabling SSH.
Router (config)# cry key generate rsa

!--- Step 3: Configure time-out and number of authentication retries.
Router (config)# ip ssh time-out 60
Router (config)# ip ssh authentication-retries 2

!--- Step 4: Configure VTYs to only accept SSH.
Router (config)# line vty 0 4
Router (config-line)# transport input ssh

!--- Step 5: Allow SSH connections only originated from the management network.
Router (config)# access-list 111 remark ACL for SSH
Router (config)# access-list 111 permit tcp 172.26.0.0 0.0.255.255 any eq 22
Router (config)# access-list 111 deny ip any any log-input
Router (config)# line vty 0 4
Router (config-line)# access-class 111 in
```

For more information on restricting which protocols are authorized on device terminal and management ports, see Restrict Device Access to Authorized Services and Protocols Only, page 2-6.

For more information about SSH configuration on IOS routers, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cg/hsec_c/part25/ch10/index.htm

Web-based GUI Access

Today almost every networking product can be configured and monitored with a Web-based user interface or GUI (graphical user interface). This type of user interface is popular because it provides a convenient way to access network equipment remotely with the use of a simple web browser.

However, some Web-based user interfaces rely on insecure protocols such as HTTP. HTTP transmits all usernames, passwords and session data in clear text. Consequently, HTTP access is vulnerable to sniffing, interception and other attacks. It is recommended that HTTP access be disabled and that Secure HTTP (HTTPS) be used as an alternative wherever possible. HTTPS uses Secure Sockets Layer (SSL) and Transport Layer Security (TLS) to provide device authentication and data encryption, delivering an acceptable level of protection.

HTTP

In Cisco IOS, HTTP is disabled by default. HTTP may be disabled using the following command:

```
Router(config)# no ip http server
```

For cases where HTTPS is not available as an alternative and HTTP access is absolutely required, the configuration guidelines outlined in Table 2-6 are recommended.

Table 2-6 HTTP Configuration

HHTP Security Guidelines	Cisco IOS Implementation
Authenticate users using AAA in conjunction with a strong password policy	`ip http authentication aaa login-authentication <aaa-listname>`
Authorize HTTP exec commands using AAA	`ip http authentication aaa exec-authorization <aaa-listname>`

Restrict incoming HTTP access attempts from a limited set of authorized HTTP management stations	`ip http access-class <ACL#>`
	`access-list <ACL#> permit host 10.0.0.1`

Limit the maximum number of concurrent HTTP connections to the expected operational number	`ip http max-connections 3`

In Cisco IOS, HTTP authentication can be enabled with the **ip http authentication** global command. The following example shows a configuration listing for HTTP authentication using TACACS+.

```
!
username adminuser privilege 15 password <mypassword>
!
aaa new-model
aaa authentication login default group adminAAAgroup local-case
aaa authorization exec default group adminAAAgroup local
aaa accounting exec default start-stop group adminAAAgroup
!
ip http server
ip http authentication aaa
!
! HTTP access requires telnet service being accepted at the VTY
line vty 0 4
 transport input telnet
```

For more information about HTTP authentication, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf005.html#wp1000960

For more details about the **http access-class** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g04.html#wp1028455

For more information about the **ip http max-connections** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g04.html#wp1028838

A sample HTTP configuration is provided in Appendix A, "Sample Configurations."

HTTPS

In Cisco IOS, HTTPS services can be enabled with the following global configuration command:

```
Router(config)# ip http secure-server
```

For more information on the **ip http secure-server** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g04.html#wp1029339

A sample HTTPS configuration is provided in Appendix A, "Sample Configurations."

SNMP Access

Simple Network Management Protocol (SNMP) is the most popular network management protocol and, as such, is widely supported in the networking industry. SNMP features support for:

- **SNMP read**

  An SNMP manager request for information.
- **SNMP write**

  An SNMP manager request to configure a device.
- **SNMP trap**

  An unsolicited notification sent from an SNMP agent, for example, an infrastructure device, to an SNMP manager.
- **SNMP inform**

  An unsolicited, acknowledged notification sent from an SNMP agent, for example, an infrastructure device, to an SNMP manager. Whilst SNMP informs are more reliable than SNMP traps, they consume more resources and are thus not typically used.

There are three versions of SNMP:

- Version 1, the oldest but still frequently supported
- Version 2c, the most commonly deployed
- Version 3, an IETF standard that provides enhanced security

SNMP versions 1 and 2c are weak in terms of security as these early versions only authenticate access to MIB objects using a community string. In addition, all communication is sent in clear text and is thus vulnerable to sniffing. Neither version supports encryption. Consequently, unauthorized users are able to execute SNMP transactions, and masquerade as legitimate users simply by sniffing and employing the configured community string. In addition, the lack of encryption facilitates the interception of SNMP messages, potentially leading to the disclosure of sensitive information.

In contrast, SNMP v3 addresses some of the security limitations of SNMP v1 and v2c by incorporating security features such as authentication, message integrity check, access controls and encryption. DES encryption enables all communication to be encrypted in order to provide confidentiality of the data. SNMP v3 also provides authorization and access controls, enabling different users to be granted different views, in accordance with the enforcement of minimum access privileges.

Consequently, it is recommended that SNMP v3, with strong authentication and encryption, should be preferred over SNMPv1 or SNMPv2c wherever it is supported due to its enhanced security features.

A summary of the security features of each version of SNMP is provided in Table 2-7.

Table 2-7 SNMP Security Features

Version	Security Level	IOS Keyword	Authentication	Encryption
SNMP v1	noAuthNoPriv	N/A	Community string	°
SNMP v2c	noAuthNoPriv	N/A	Community string	°
SNMP v3	noAuthNoPriv	noauth	Username	°
	authNoPriv	auth	MD5 or SHA	°
	authPriv	priv	MD5 or SHA	DES-56

The general guideline for securing SNMP access is to disable SNMP access if it is not required. If SNMP access is required, it is recommended that the following configuration guidelines be considered:

- Selectively use SNMP for required actions only
- Restrict actions to read-only queries

Note Write access creates significant risk and is not recommended.

- Deny queries that request to download the full IP routing and ARP tables using SNMP views
- Treat community strings like root passwords
- Delete default community strings
- Define strong, non-trivial community strings
- Restrict incoming SNMP access attempts to a limited set of authorized SNMP management stations through the use of extended ACLs using recognized SNMP ports (UDP 161 and 162)
- Restrict the SNMP actions permitted by any particular SNMP management station by using different SNMP community strings with different associated SNMP views and different ACLs, permitting only the minimal required level of information to any particular SNMP management station
- If only SNMP v3 is being used, ensure only SNMP v3 access is enabled and with the highest possible security level supported by the communicators
- Enable only operationally important traps (e.g., BGP state changes)
- Send a trap on community name authentication failures to track failed access attempts
- Send a trap for configuration changes and environmental monitor threshold exceptions
- Ensure SNMP traps are regularly monitored

If SNMP v1 or SNMP v2c must be used, it is recommended that IPSec or an out-of-band (OOB) management network are considered as a means of protecting SNMP v1 and SNMP v2c transactions from attack.

A sample SNMP configuration is available in Appendix A, "Sample Configurations."

For more information about SNMPv3, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf014.html

Locally Stored Information Protection

Cisco IOS devices store some sensitive information locally, including passwords and secrets. Passwords should generally be maintained and controlled by a centralized AAA server. Centralized AAA servers are preferred as they provide a number of features that facilitate secure password management, including the ability to enforce strong passwords, force users to periodically change their passwords, lock accounts after a specific number of failed login attempts, and many other useful options. In addition, they enable the passwords to be stored and backed up in a secure manner.

However, even if a centralized AAA server is deployed, some locally stored passwords are required on for certain cases, such as local fallback in the case of AAA servers not being available, special-use usernames, secret keys, and other password information. These local passwords and other sensitive information are stored locally in the configuration file.

Cisco IOS offers the following features to enable locally stored sensitive information to be retained in a secure manner in the configuration file:

- Global password encryption
- Local user password encryption
- Enable secret

It should be noted that the encryption of locally stored passwords is employed to preserve the confidentiality of sensitive information stored in a configuration file, both during local viewing of the file and during its transfer. Local password encryption does not address the vulnerability of passwords and secrets to dictionary attacks. See the section Restrict Login Vulnerability to Dictionary and DoS Attacks for more information.

Global Password Encryption

In a Cisco IOS configuration file or listing, by default, some passwords and secrets are stored and presented in clear text. Local passwords and secrets stored in an IOS configuration file should be encrypted to prevent over-the-shoulder browsing of sensitive information.

Cisco IOS offers the ability to encrypt locally stored passwords, CHAP secrets, and similar data in the configuration file. This is enabled using the following global configuration command:

```
Router(config)# service password-encryption
```

However, the encryption algorithm used by the **service password-encryption** command is a simple Vigenere cipher (Type 7) that can be easily reversed. Consequently, this command is primarily only useful for protection from *shoulder surfing*, where an unauthorized individuals attempts to view passwords in a configuration file simply by looking over the shoulder of an authorized user.

Note that not all sensitive data in a configuration file is encrypted by this command. Consequently, configuration files should always be treated as sensitive documents and properly protected to ensure the confidentiality of the data. Particular concern and necessary steps should be taken to protect them during transfer.

Cisco IOS offers support for a stronger encryption algorithm (Type 5) for some locally stored passwords and this should be leveraged whenever available. For example, define local users using the **secret** keyword instead of the **password** keyword, and use **enable secret** instead of **enable password**.

For more information on the **service password-encryption** command, refer to the following URL :

http://www.cisco.com/en/US/docs/ios/12_3/security/command/reference/sec_r1g.html#wp1070450

Local User Password Encryption

Local user names and passwords should be stored in the most secure manner available on a device. These local accounts should only be used as a local fallback method in case the AAA servers for AAA-based authentication are not available.

In Cisco IOS, the passwords for locally configured usernames can be stored using a strong, MD5 encryption algorithm (Type 5) by using the secret keyword instead of the password keyword, where available. For example:

```
Router(config)# username <name> secret <strongpassword>
```

Note that MD5 encryption is not retrievable and thus cannot be used with protocols that require clear text passwords, such as Challenge Handshake Authentication Protocol (CHAP).

For more information on the **username** <*name*> **secret** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_t2.html#wp1031804

Enable Secret

Local privileged level administrative passwords should be stored in the most secure manner available on a device. Privileged level access typically refers to a level of access which provides the ability to configure a network infrastructure device. Local privileged level administrative passwords should only be used as a local fallback method in case the AAA servers for AAA-based authentication are not available.

In Cisco IOS, privileged level EXEC sessions are accessed using the **enable** command. It is recommended that an **enable secret** be configured instead of an **enable password**, as the **enable secret** uses a strong, MD5 encryption algorithm (Type 5) which is not reversible.

To set an **enable secret**, use the global configuration command:

```
Router(config)# enable secret <strongpassword>
```

By default, an enable secret password is not configured and as a general best practice, one should always be set. If no enable secret is set and a password is configured for the console TTY line, the console password can be used to obtain privileged level access, even from a remote VTY session. Consequently, it is recommended that an enable secret always be configured on a device.

If an enable password is configured, it should be removed to ensure that enable access can only be obtained using the enable secret. An enable password can be removed with the global configuration command:

```
Router(config)# no enable password
```

It should be noted that privileged level access remains vulnerable to dictionary attacks even if an enable secret is employed. For more information on how to address dictionary attacks, see the section Restrict Login Vulnerability to Dictionary and DoS Attacks.

For more information about the **enable secret** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_e1.html#wp1022924

Infrastructure Device Management Access Logging

It is critical to ensure that infrastructure device access and configuration changes are logged to record the following information:

- Who accessed a device
- When a user logged in
- What a user did
- When a user logged off
- Failed access attempts
- Failed authentication requests
- Failed authorization requests

This information is invaluable for forensic analysis in the case of unauthorized attempts or access, as well as for configuration change issues and to help plan group administration changes. It may also be used in real time to identify anomalous activity which may indicate that an attack is taking place. Automated tools are available to enable this analysis and correlate information from additional sources, such as IDS and firewall logs.

The baseline logging objectives and the Cisco IOS features available to achieve them are outlined in Table 2-8.

Table 2-8 Features for Infrastructure Device Management Access Logging

Baseline Logging Objectives	Available Cisco IOS Features
Log successful device access attempts	AAA EXEC accounting
	Syslog login success notification [1]
Log failed device access attempts	AAA failed authentication accounting
	Syslog login failure notification[2]
Log commands entered in EXEC and privilege modes	AAA command accounting
	Archive configuration change logger
Log system-level events, such as system reboots or accounting on/off	AAA system accounting

[1] A syslog notification of login success or failure can be used to complement the AAA EXEC accounting records.

AAA EXEC Accounting

Cisco IOS AAA EXEC accounting provides information about user EXEC sessions on the network infrastructure device, including username, date, start and stop times, the device IP address ,and the user source IP address.

A syslog notification of login success or failure can be used to complement the AAA EXEC accounting records.

The following is an example of TACACS+ EXEC accounting records generated for an administrative session. These accounting records were extracted from the TACACS+ Accounting active log of a Cisco ACS server.

```
Date          Time        User-Name     Group-Name Caller-Id       Acct-Flags elapsed_time service   task_id NAS-Portname  NAS-IP-Address
     2/25/2008     17:11:20 admin2      Group 1    172.26.158.234  start                   shell          20 tty1          172.26.158.235
     2/25/2008     17:11:41 admin2      Group 1    172.26.158.234  stop                 21 shell          20 tty1          172.26.158.235
```

AAA EXEC accounting is enabled with the aaa accounting exec command. A sample configuration using a named method list and AAA server group, plus its enforcement on VTY lines is shown below:

```
!
aaa accounting exec account-exec-list start-stop group adminAAAgroup
!
line vty 0 4
 accounting exec account-exec-list
!
```

For more information on the **aaa accounting** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_a1.html#wp1038916

For more information on AAA, see AAA Services, page 2-14.

AAA Failed Authentication Accounting

When AAA accounting is enabled, Cisco IOS software does not generate accounting records for system users who fail login authentication, or who succeed in login authentication but fail PPP negotiation for some reason.

To specify that accounting stop records be generated for users who fail authentication at login or fail PPP session negotiation, use the following command in global configuration mode:

```
Router (config)# aaa accounting send stop-record authentication failure
```

Note This feature does not currently permit the definition of a named AAA method list or a AAA server group. The first RADIUS server configured on the device using **radius-server host** *name* command is the one to which accounting records will be sent.

For more information on the **aaa accounting send stop-record authentication failure** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_a1.html#wp1040064

For more information on AAA, see the AAA Services, page 2-14.

AAA Command Accounting

Available only with TACACS+, Cisco IOS AAA Command accounting provides information about the EXEC session commands for a specified privilege level that are executed on a network infrastructure device. Each command accounting record includes a list of the commands executed for that privilege level, the date and time each command was executed and the user who executed it.

The following is an example of TACACS+ command accounting records generated for an administrative session. These accounting records were extracted from the TACACS+ Administration active log of a Cisco ACS server.

```
Date        Time      User-Name  Group-Name  cmd                          priv-lvl  service  NAS-Portn task_id  NAS-IP-Address
 2/25/2008   17:40:39 admin2      Group 1     configure terminal<cr>              15 shell    tty1           33 172.26.158.235
 2/25/2008   17:40:55 admin2      Group 1     hostname c19-6500-4 <cr>            15 shell    tty1           34 172.26.158.235
 2/25/2008   17:40:58 admin2      Group 1     write <cr>                          15 shell    tty1           35 172.26.158.235
```

AAA command accounting is enabled with the following global configuration command:

```
aaa accounting exec
```

A sample configuration using a named method list and AAA server group, plus its enforcement on VTY lines is shown below:

```
!
aaa accounting commands 15 account-exec-list start-stop group tacacs-group
!
line vty 0 4
 accounting commands 15 account-exec-list
!
```

Note When **command accounting** is enabled, *all* commands entered in enable mode will be logged in the accounting records on the accounting host. Consequently, any changes to sensitive information on a device, such as an enable secret, should *not* be entered on the CLI, unless command accounting is temporarily disabled. The recommended approach is to update the configuration offline and securely download the new configuration to the device.

For more information on the **aaa accounting** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_a1.html#wp1038916

For more information on AAA, see AAA Services, page 2-14.

AAA System Accounting

System accounting provides information about system-level events not associated with users, including when the system reboots or when accounting is turned on or off.

In Cisco IOS, AAA system accounting is enabled with the following command:

```
aaa accounting system
```

A sample configuration enabling system accounting for start and stop events to a AAA server group is shown below.

```
Router (config)# aaa accounting system default start-stop group tacacs-group
```

System accounting does not support named accounting but does support named AAA server groups.

A sample system accounting record indicating that AAA accounting has been turned off is shown below.

```
Mon Feb 25 17:13:42 2008        172.26.158.234    unknown unknown unknown start
task_id=25  service=system  event=sys_acct  reason=reconfigure
```

Syslog Login Success and Failure Notifications

Cisco IOS offers the ability to send a syslog trap upon login success or login failure and can be used to complement the AAA EXEC accounting records.

This feature is one of the IOS login enhancements introduced in Cisco IOS Release 12.3(4)T and 12.2(25)S.

The generation of syslog traps for successful and failed login attempts is enabled using the following commands respectively.

```
Router(config)# login on-success log
Router(config)# login on-failure log
```

A sample syslog message for a successful login is shown below:

```
Sep 25 12:49:32.465 UTC: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin2
] [Source: 172.26.158.234] [localport: 22] at 12:49:32 UTC Thu Sep 25 2003
```

A sample syslog message for a failed login attempt is shown below:

```
Sep 25 13:19:46.864 UTC: %SEC_LOGIN-4-LOGIN_FAILED: Login failed [user: ] [Sourc
e: 172.26.158.234] [localport: 22] [Reason: Login Authentication Failed] at 13:1
9:46 UTC Thu Sep 25 2003
```

For more information on the **login on-success** and **login on-failure** commands, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_k1.html#wp1031689

Configuration Change Notification and Logging

Cisco IOS offers a Configuration Change Notification and Logging (Config Log Archive) feature which tracks commands executed in configuration mode through the CLI or HTTP. A syslog notification of configuration changes may also be enabled.

The log record includes the following information:

- Configuration command that was executed
- Configuration mode in which the command was executed
- User who executed the command
- Time at which the command was executed
- Configuration change sequence number
- Parser return codes for the command

For more information on the Configuration Change Notification and Logging (Config Log Archive) feature, refer to the following URL:

http://www.cisco.com/en/US/partner/products/ps6350/products_configuration_guide_chapter09186a0080454f73.html

Archive configuration change logging is enabled as shown in the following example:

```
! Enter archive mode
Router(config)# archive
!
! Enter the archive configuration change logger configuration mode
Router(config-archive)# log config
!
```

```
! Enable configuration change logging
Router(config-archive-log-config)# logging enable
!
! Set the maximum number of configuration change log entries as 200
Router(config-archive-log-config)# logging size 200
!
! Prevent passwords from being displayed in the configuration log
Router(config-archive-log-config)# hidekeys
!
! Enable configuration change messages to be sent to a syslog server
Router(config-archive-log-config)# notify syslog
!
```

For more information on the **archive** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_a1.html#wp1018716

Displaying Configuration Change Log Entries

The configuration change log may be viewed using the command **show archive log config** command as follows:

```
Router# show archive log config all
```

http://www.cisco.com/en/US/partner/docs/ios/12_4/cfg_fund/command/reference/cfn_07h.html#wp1034746

A line-by-line comparison of a specified configuration file to the running configuration file, which generates a list of the configuration lines that do not appear in the running configuration file, can be viewed using the command **show archive config incremental-diffs** command, refer to the following URL:

```
Router# show archive config incremental-diffs nvram:startup-config
```

http://www.cisco.com/en/US/partner/docs/ios/12_4/cfg_fund/command/reference/cfn_07h.html#wp1039115

A line-by-line comparison of any two configuration files (accessible through the Cisco IOS File System [IFS]) and a list of the differences between them can be generated with the **show archive config differences** command as follows:

```
Router# show archive config differences nvram:startup-config
```

For more information on the **show archive log** command, refer to the following URL:

http://www.cisco.com/en/US/partner/products/ps6350/products_configuration_guide_chapter09186a0080454f73.html

For more information on the **show archive config incremental-diffs** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_s1.html#wp1077902

For more information on the **show archive config differences** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_s1.html#wp1075054

General Device Access and Configuration Change Logging Best Common Practices

General device access and configuration change logging best common practices include:

- Logs should be written to a secure server through a secure, reliable path.
- Ensure that the logs and the servers themselves are properly secured.
- Logged information should be reviewed on a regular basis.
- A high rate of failed authentication attempts may indicate password guessing.
- A high rate of failed authorization attempts may indicate device compromise.
- Only required information should be logged to minimize the amount of traffic generated and the post-processing required.
- It is critical to ensure that NTP is employed to generate logs with consistent, synchronized timestamps across the entire network infrastructure. For more information, refer to Time Synchronization, page 5-2.

File Transfer

Files which may need to be transferred to network infrastructure devices include image and configuration files. The secure transfer of these files is critical to network infrastructure security.

Common file transfer techniques include:

- File Transfer Protocol (FTP)
- Trivial File Transfer Program (TFTP)
- Secure Copy (SCP)

Note Both FTP and TFTP offer minimal security and transfer data in clear text, whereas SCP uses SSH for authentication and encryption. Consequently, SCP should be used wherever feasible.

File Transfer Protocol (FTP)

FTP requires a username and password in order to authenticate access to the FTP server, however all data is sent in clear text. Consequently, though FTP offers a minor advantage over using TFTP to upload a configuration file to a server, it remains vulnerable to sniffing of both the username and password, as well as the configuration file itself. This may be of significant concern if one considers the sensitive data within a configuration file.

It is recommended that Secure Copy (SCP) be used as an alternative to FTP wherever feasible. If FTP must be used, it is recommended that transactions be conducted from a loopback or out-of-band (OOB) interface on a device, thereby enabling access to the FTP server to be restricted to authorized source IP addresses.

```
Router(config)# ip ftp source-interface <Loopback-OOB>
```

Source IP spoofing protection should also be deployed.

It is recommended that the FTP username and password are not defined in the configuration file itself but entered as part of the copy command.

The use of IPSec between a device and the FTP server itself, or perhaps its network subnet, should be considered for transfers over public networks.

Trivial File Transfer Program (TFTP)

TFTP transactions are not authenticated and all data is sent in clear text. Consequently, TFTP is vulnerable to unauthorized access, as well as sniffing. This may be of significant concern if one considers the sensitive data within a configuration file.

It is recommended that Secure Copy (SCP) be used as an alternative to TFTP wherever feasible. If TFTP must be used, it is recommended that transactions be conducted from a loopback or out-of-band (OOB) interface on a device, thereby enabling access to the TFTP server to be restricted to authorized source IP addresses.

```
Router(config)# ip tftp source-interface <Loopback-OOB>
```

Source IP spoofing protection should also be deployed.

The use of IPSec between a device and the TFTP server itself, or perhaps its network subnet, should be considered for transfers over public networks.

Secure Copy (SCP)

SCP relies on SSH for secure authentication and transport, enabling the secure and authenticated copying of files. Consequently, SSH must be already configured on a device in order for SCP to be enabled. See the Secure Shell (SSH) section for full details on how to enable SSH.

In Cisco IOS, SCP can be enabled with the following global configuration command:

```
Router(config)# ip scp server enable
```

A secure copy is initiated through the standard **copy** command. SCP is available as one of the transfer options. For example:

```
cr20-6500-4#copy disk1:remote.cfg scp://admin2@172.26.159.164
Address or name of remote host [172.26.159.164]?
Destination username [admin2]?
Destination filename [remote.cfg]?
Writing remote.cfg
Password:
!!
6258 bytes copied in 7.192 secs (870 bytes/sec)
cr20-6500-4#
!
```

AAA-based authentication and authorization should be employed to ensure only authorized users are able to perform an SCP operation.

```
! AAA authentication and authorization must be configured properly for SCP to work.
aaa new-model
aaa authorization exec default group tacacs-group local
aaa authorization exec default group tacacs-group local
username <admin-user> privilege 15 password <password>
! SSH must be configured and functioning properly.
ip ssh time-out 120
ip ssh authentication-retries 3
```

For more information on the **ip scp server enable** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_i2.html#wp1031870

Device Software Image Verification

The software installed and running on a network infrastructure device should be verified to ensure that it is valid. Cisco IOS offers an image verification feature which validates the MD5 digest of an IOS image. If available, digital signing of software should be leveraged to address authentication and non-repudiation issues.

IOS Software Image Verification

The Cisco IOS software image verification feature provides a user-friendly mechanism for validating the MD5 digest of an IOS image. This may be configured to occur automatically, upon any ‘copy’ or ‘reload’, or it may be enabled as a manual option when entering the ‘copy’ or ‘reload’ command. A verification may also be initiated from the CLI.

Image verification is not currently supported on non-IOS image files.

To enable automatic image verification, use the Cisco IOS command:

```
Router(config)# file verify auto
```

Manual image verification may be initiated from the CLI using the `verify` command:

```
Router# verify location://image
```

Manual image verification may be initiated from the CLI during a copy:

```
Router# copy /verify <source> <dest>
```

Manual image verification may also be initiated from the CLI during a reload:

```
Router# reload /verify
```

For more information on the Image Verification feature, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_image_verifctn_ps6350_TSD_Products_Configuration_Guide_Chapter.html

Infrastructure Management Network

An infrastructure management network refers to the network carrying control and management plane traffic (such as NTP, SSH, Telnet, SNMP, syslog, FTP, etc) for the infrastructure devices themselves. Device access can be through the console, as well as through Ethernet or other network management interfaces.

This control and management plane traffic is critical to network operations, providing visibility into and control over the network. Consequently, a well-designed and secure infrastructure management network is critical to the overall security and operation of a network.

One of the key recommendations for a secure infrastructure management network is the separation of management and data traffic in order to ensure remote manageability even under high load and traffic conditions.

Infrastructure management network implementation approaches include:

- Out-of-band (OOB)

  An OOB management network consists of a network which is completely independent and physically disparate from the data network that it helps to manage. This is also sometimes referred to as a Data Communications Network (DCN). Network devices may connect to the OOB network in different ways:

  - Some network devices feature built-in management interfaces that may be used to connect to the OOB network.
  - Some platforms allow the configuration of physical interfaces as dedicated management interfaces. For example, in Cisco IOS Management Plane Protection restricts management traffic to predefined management interfaces.
  - Network devices may also connect to the OOB network with dedicated data interfaces. In this case, ACLs should be deployed to ensure management traffic is only handled by the dedicated interfaces.

- **Pseudo out-of-band**

  A pseudo out-of-band management network uses the same physical infrastructure as the data network but provides logical separation through the virtual separation of traffic, such as using VLANs or VPNs.

- **In-band**

  An in-band management network uses the same physical and logical paths as the data traffic.

An OOB management network provides the maximum visibility and control as it is not impacted by incidents on the data network. An OOB management network does require the deployment and management of separate infrastructure, including, perhaps, redundancy for maximum availability. However, management network traffic does not typically demand high bandwidth or high performance devices and so the costs are typically reasonable. In addition, the number of devices required is typically small, requiring only sufficient density to provide connectivity to each infrastructure device being managed.

Ultimately, the design decision requires a per-customer analysis of risk versus benefits and costs.

Some general considerations include:

- An isolated OOB management network maximizes visibility and control over the network even during disruptive events.
- Transmitting network telemetry over an OOB network minimizes the chance for disruption of the very information which provides critical network visibility.
- In-band management access to network infrastructure, hosts, etc, is vulnerable to complete loss in the event of a network incident, removing all network visibility and control. Appropriate QoS controls should be put in place to mitigate this.
- Many network infrastructure devices feature interfaces which may be dedicated to device management, including serial console ports and Ethernet management interfaces.
- An OOB management network can typically be deployed at a reasonable cost, since management network traffic does not typically demand high bandwidth, nor high performance, devices and only requires port density sufficient to support connectivity to each infrastructure device.

Device Management Best Common Practices

- Enforce a strong password policy
- Assign and enforce per-user accounts

  Each user should have an individual, unique username and password
- Remove default accounts
- Change default passwords
- Grant minimum access privileges
- Force users to periodically change their passwords
- Selectively use SNMP and treat community strings like root passwords
- Employ secure management protocols where available, including SSH, SCP, SSL, OTP etc.
- If insecure management protocols such as Telnet, syslog, SNMP, TFTP, FTP, etc. are required, consider out- of-band (OOB) management
- If OOB management is not possible, restrict access to the management protocols using the “set ip permit” lists on the management protocols
- Put the management VLAN into a dedicated non-standard VLAN where nothing but management traffic resides and consider physically back-hauling this interface to the management network
- Review the password recovery settings

CHAPTER 3

Routing Infrastructure

Routing is one of the most important parts of the infrastructure that keeps a network running, and as such, it is absolutely critical to take the necessary measures to secure it. There are different ways routing can be compromised, from the injection of illegitimate updates to DoS specially designed to disrupt routing. Attacks may target the router devices, the peering sessions, and/or the routing information. Fortunately, protocols like BGP, IS-IS, OSPF, EIGRP and RIPv2 provide a set of tools that help secure the routing infrastructure. This section provides the guidelines for using such tools.

The router's primary functions are to learn and propagate route information, and ultimately to forward packets via the most appropriate paths. Successful attacks against routers are those able affect or disrupt one or more of those primary functions by compromising the router itself, its peering sessions, and/or the routing information.

Routers are subject to the same sort of attacks designed to compromise hosts and servers, such as password cracking, privilege escalation, buffer overflows, and even social engineering. Most of the best practices in this document help mitigate and even prevent some of those threats.

Peering relationships are also target of attacks. For most routing protocols routers cannot exchange route information unless they establish a peering relationship, also called neighbor adjacency. Some attacks attempt to break established sessions by sending the router malformed packets, resetting TCP connections, consuming the router resources, etc. Attacks may also prevent neighbor adjacencies from being formed by saturating queues, memory, CPU and other router resources. This section of the document presents a series of best practices to protect neighbor adjacencies from those threats.

Finally, routing can also be compromised by the injection of false route information, and by the modification or removal of legitimate route information. Route information can be injected or altered by many means, ranging from the insertion of individual false route updates to the installation of bogus routers into the routing infrastructure. Potential denial of service conditions may result from intentional loops or black-holes for particular destinations. Attackers may also attempt to redirect traffic along insecure paths to intercept and modify user's data, or simply to circumvent security controls. This section also includes a collection of best practices designed to prevent the compromising of routing information.

CSF Methodology Assessment

The results of applying the CSF methodology are shown inTable 3-1 and Table 3-2. The tables highlight the technologies and features identified for securing the routing plane and which are integrated in Network Secure Baseline.

Total Visibility

Table 3-1 Routing Infrastructure—Total Visibility

Identify	Monitor	Correlate
• Neighbor Authentication • Routing Peer Definition • Route Redistribution Filtering	• Neighbor adjacency logging • Logging – Syslog – SNMP	

Complete Control

Table 3-2 Routing Infrastructure—Complete Control

Harden	Isolate	Enforce
• Default Passive Interface • BGP TTL Security Check • Standby interfaces • Standby devices • Element redundancy • Topological Redundancy		• Prefix Filtering • Maximum Prefix Filtering • Route Redistribution Filtering • Stub Routing • iACL • rACL • Control Plane Protection • Control Plane Policing

Restricted Routing Protocol Membership

Many dynamic routing protocols, particularly interior gateway protocols, implement automatic peer discovery mechanisms that facilitate the deployment and setup of routers. By default, these mechanisms operate under the assumption that all peers are to be trusted, making it possible to establish peering sessions from bogus routers and to inject false routing data. Fortunately, Cisco IOS provides a series of recommended features designed to restrict routing sessions to trusted peers and that help validate the origin and integrity of routing updates:

- Neighbor authentication
- Routing peer definition
- Default passive interface
- BGP TTL Security Check
- iACLs
- rACLs
- Control Plane Policing
- Control Plane Protection

Neighbor Authentication

Neighbor authentication is a feature available on most routing protocols, and that ensures a router only receives reliable routing information and from trusted neighbors. That is achieved by certifying the authenticity of each neighbor and the integrity of its routing updates. Technically, each router is initially configured with a shared secret key that is used to validate each routing update. Before sending a routing update, each router is required to sign it with the predefined secret key; and include the resulting signature as part of the update message. Finally, the update is verified by the receiving neighbor to prove its authenticity and integrity. Neighbor authentication is supported for BGP, IS-IS, OSPF, RIPv2 and EIGRP.

Neighbor authentication helps protect peering sessions from attacks such as session reset attempts and insertion of unauthorized routing peers. Neighbor authentication also helps secure routing data from the injection of false routes, and the removal or modification of legitimate routing information from unauthorized routing peers. It should be noted however that neighbor authentication does not prevent incorrect routing data from being injected by a valid router that has been compromised trusted router. Fortunately, such attack scenarios can be mitigated by route filtering, as explained later in this section.

Most routing protocols support two types of neighbor authentication, plain text and Message Digest Algorithm Version 5 (MD5) authentication. Plain text authentication consists on sending the secret key in the clear inside each routing update message, which does not provide much security since keys can be intercepted while in transit. MD5 authentication works by processing each routing update with a MD5 hash function; and by including the resulting signature (digest) as part of the routing update message. This method is more secure because the actual shared secret key is never sent over the network. For this reason, MD5 authentication should be preferred over clear text.

Enabling neighbor authentication is a recommended practice for all routers, but especially for those more exposed to threats such the routers facing the Internet or other external networks. Ideally, secret keys should be unique to each peering relationship or interface, in the case of broadcast media like Ethernet. However, having all unique passwords may pose an operational challenge in large networks; hence, it is up to the administrators to find the right balance between security and the easy of operation.

The following example shows the configuration of OSPF MD5 neighbor authentication on an IOS router:

```
! OSPF MD5 authentication
interface Ethernet1
  ip address 10.139.20.1 255.255.255.0
  ip ospf message-digest-key 10 md5 oursharedsecret
!
router ospf 20
  network 10.139.20.0 0.0.0.255 area 0
  area 0 authentication message-digest
```

In EIGRP MD5, authentication is enabled at the interface or subinterface level as shown in the following example. Note that once EIGRP MD5 authentication is enabled on an interface or subinterface, the router stops processing routing messages received from that interface or subinterface until the peers are also configured for message authentication. This does interrupt routing communications on your network.

```
! EIGRP authentication
interface Ethernet 1
 ip authentication mode eigrp 10 md5
 ip authentication key-chain eigrp 10 mychain
!
router eigrp 10
 network 10.0.0.0
!
key chain mychain
 key 1
  key-string oursharedsecret
!
```

Similar to EIGRP, in RIP version 2 authentication is enabled at the interface or subinterface level as shown in the following example. Note that once RIPv2 MD5 authentication is enabled on an interface or subinterface, the router stops processing routing messages received from that interface or subinterface until the peers are also configured for message authentication. This does interrupt routing communications on your network.

```
interface ethernet 0
 ip rip authentication key-chain mychain
 ip rip authentication mode md5
!
router rip
 network 10.0.0.0
 version 2
!
key chain mychain
 key 1
 key-string oursharedsecret
!
```

In BGP MD5 neighbor authentication is configured within the routing process and as a neighbor attribute, as shown in the example below. Note that once BGP MD5 authentication is enabled for a peer, no peering session will be established until the peer is also configured for message authentication. This does interrupt routing communications on your network.

```
router bgp 10
  no synchronization
  bgp log-neighbor-changes
  network 64.104.0.0
  neighbor 198.133.219.10 remote-as 10
  neighbor 198.133.219.10 password 7 05080F1C22431F5B4A
!
```

Refer to the *Cisco IOS IP Routing Protocols Configuration Guide* fat the following URL for your IOS Software Release to learn more on how to configure neighbor authentication in your routing protocol.

http://www.cisco.com/web/psa/products/tsd_products_support_configure.html

Routing Peer Definition

The same dynamic peer discovery mechanisms that facilitate the deployment and setup of routers can be used potentially to insert bogus routers into the routing infrastructure. This problem may be prevented by disabling such mechanisms by statically configuring a list of trusted neighbors with known IP addresses. This can be used in conjunction with other routing security features such neighbor authentication and route filtering.

In the case of EIGRP, by default, most hello, update, and query messages are sent in multicast packets on broadcast interfaces like Ethernet. This default behavior can be changed by configuring a static neighbor, after which all routing messages are sent in unicast packets. In addition, after the first static neighbor is configured, the router only accepts EIGRP packets from peers that are explicitly configured with a neighbor statement. Consequently, any messages coming from routers without a corresponding neighbor statement are discarded.

The fact the router discards any messages coming from any routers not expressly configured as neighbors helps prevent the insertion of unauthorized routing peers. At the same time, the change of routing messages from multicast to unicast makes unauthorized interception of routing data more difficult, preventing a potential attacker from learning topological information that could be used to carry out future attacks. It should be noted however that, since static neighbors are recognized by their

IP addresses, there is a possibility of IP address spoofing on these neighbors. With the adequate knowledge, an attacker may spoof the IP address of a valid static neighbor. Neighbor authentication provides a second layer of protection to help mitigate IP spoofing. With neighbor authentication, sessions and updates are only accepted from neighbors that use the proper secret keys. As a result, a spoofing attempt will not succeed as long as the secret keys are unknown to the attacker.

It should be also noted that static neighbors do not prevent incorrect routing data from being injected by a compromised trusted router. Fortunately, such attack scenarios can be mitigated by route filtering, as later explained in this section.

The definition of static neighbors is a recommended practice for all routers, but especially for those at the Internet edge or facing other external networks.

As an example, EIGRP static peers can be defined by using the **neighbor** command. Dynamic peer discovery is disabled as soon as the first static neighbor is configured. The following example configures EIGRP peering sessions with the 192.168.1.1 and 192.168.2.2 neighbors:

```
router eigrp 100
 network 10.0.0.0
 neighbor 10.139.20.1 FastEthernet0/0
```

Note In OSPF the configuration of static neighbors does not prevent other routers in the network from establishing an adjacency. For this reason, the technique here described does not apply for OSPF.

Refer to the *Cisco IOS IP Routing Protocols Configuration Guide* fat the following URL for your IOS Software Release to learn how to configure static peers with your routing protocol.

http://www.cisco.com/web/psa/products/tsd_products_support_configure.html

Default Passive Interface

In large service provider and enterprise networks some routers often have a large number of interfaces, for example, at the WAN edge. A common practice to facilitate the configuration of a routing protocol on such routers is to enable the routing processes on a network range matching all the interfaces. While this technique facilitates the configuration of the routing protocol, enabling routing indiscriminately on all interfaces may increase the chances for the insertion of unauthorized routing peers. To prevent those problems, one can either manually set the **passive-interface** command on the interfaces where router adjacencies are not expected, or set all interfaces as passive by default with the router **passive-interface default** command. The **passive-interface default** command changes the configuration logic to a default passive, therefore interfaces where router adjacencies are expected need to be configured with the **no passive-interface** command. This feature works for all routing protocols that support the **passive-interface** command, and has been tested with all supported Cisco routing protocols.

It should be noted that the effects of the **passive-interface** vary depending on the routing protocol. In RIP and IGRP, the **passive-interface** command stops the router from sending updates on the selected interface, but the router continues listening and processing updates received from neighbors on that interface. In EIGRP and OSPF, the **passive-interface** command prevents neighbor sessions to be established on the selected interface. This stops not only routing updates from being advertised, but it also suppresses incoming routing updates.

In terms of security, using the **passive-interface** command in RIP and IGRP helps control the propagation of routing updates, but it does neither prevent the insertion of unauthorized peers nor the manipulation of incoming routing updates. On the other hand, configuring an interface as passive in EIGRP and OSPF prevents the establishment of peering relationships on that interface, therefore preventing unauthorized peering sessions as well as the insertion, modification, and/or removal of routing information. At the same time, passive interfaces protect the EIGRP and OSPF routers from routing–based DoS attacks that may arrive from the passive interfaces.

To summarize, using the **passive-interface default** command is a safe practice when enabling EIGRP or OSPF on a range matching multiple interfaces. In the case of RIP or IGRP, the **passive-interface default** command does not protect the router from unauthorized peers or the manipulation of incoming routing updates. For this reason, it is not recommended to enable these protocols on network ranges matching interfaces that are to be passive.

In the following example, all interfaces running OSPF are set to passive, while only interface serial 0 is enabled. This means that neighbor adjacencies may only be formed with peers reachable through interface serial 0:

```
router ospf 100
 passive-interface default
 no passive-interface Serial0
 network 10.139.5.0 0.0.0.255 area 0
 network 10.139.20.0 0.0.0.255 area 4
```

In this example, all interfaces running EIGRP are configured as passive, while the serial 0 interface is enabled:

```
router eigrp 10
 passive-interface default
 no passive-interface Serial0
 network 10.0.0.0
```

In this example, RIP is configured to not send any routing information on all interfaces except via interface serial 0. With this configuration, however, any incoming routing updates from any interface will continue to be received and processed:

```
router rip
 passive-interface default
 no passive-interface Serial0
 network 10.0.0.0
 version 2
```

More information on the **no passive-interface default** command can be found on the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1830/products_feature_guide09186a008008784e.html

BGP TTL Security Check

TTL Security Check is a security feature that protects BGP peers from multi-hop attacks. This feature is based on the Generalized TTL Security Mechanism (GTSM, RFC 3682), and is currently available for BGP. Work is currently in progress to implement this feature for other routing protocols such as OSPF and EIGRP.

TTL Security Check allows the configuration of a minimum acceptable TTL value for the packets exchanged between two eBGP peers. When enabled, both peering routers transmit all their traffic to each other with a TTL of 255. In addition, routers establish a peering session only if the other eBGP peer sends packets with a TTL equal to or greater than the TTL value configured for the peering session. All packets received with TTL values less than the predefined value are silently discarded.

TTL Security Check prevents routing-based DoS attacks, unauthorized peering and session reset attacks launched from systems not directly connected to the same subnet as the victim routers. Though it should be noted that TTL Security Check does not provide integrity or authentication between BGP peers, and it does neither stop attacks launched from already compromised routers.

Because TTL Security Check provides protection against multi-hop attacks, it is recommended to enable it on all BGP routers, but especially on those in contact with external peers.

In Cisco IOS software, the TTL security check can be enabled per peer with the **neighbor ttl-security** command:

```
Router(config)# router bgp as-number
Router(config-router)# neighbor ip-address ttl-security hops hop-count
```

In this example, TTL security check is enabled for the 10.1.1.1 eBGP neighbor, and which resides two hops away:

```
Router(config)# router bgp 10
Router(config-router)# neighbor 198.133.219.10 ttl-security hops 2
```

For more information about TTL Security Check, refer to the following URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a0080455621.html

iACLs

Infrastructure protection access control lists (iACLs), are extended ACLs designed to protect the routing and switching infrastructure by only permitting legitimate routing and management traffic that originates from authorized devices. iACLs may be deployed as the first line of defense against external threats, therefore they may be configured at network ingress points such as the enterprise Internet edge, or the SP peering points. iACLs may be also implemented as a barrier to protect inner network segments from internal threats. In terms of routing, by only allowing routing sessions and traffic from valid peers, iACLs help protect routers from unauthorized access and DoS attacks based on unauthorized protocols and sources. In addition, they protect routing sessions by preventing the establishment of unauthorized sessions, and by reducing the chances for session reset attacks. iACLs however, are not effective mitigating attacks sourced from trusted routers and based on trusted protocols. iACLs are explained in more detail in Chapter 4, "Device Resiliency and Survivability."

rACLs

Receive ACL (rACL) is a feature available on some Cisco distributed router platforms that allows to filter traffic destined to the central Route Processor (RP) at the line card level, and before it reaches the RP. This protects the router from attacks designed to overwhelm the capacity of the RP. A rACL consists of a standard or an extended ACL that is first created on the RP, and that is then pushed to the line card CPUs. When a packet destined to the RP enters a line card, the line card CPU processes the packet against the rACL, and it forwards the packet to the RP only if permitted by the rACL. rACL is currently supported on Cisco 1200 Series Routers, Cisco 7500 Series Routers and Cisco 10720 Internet Routers.

By enforcing an ACL at the line cards and by blocking unwanted traffic before it reaches the RP, rACLs help mitigate a wide range of DoS attacks based on control plane traffic, as well as unauthorized access attempts. In addition, rACLs protect routing sessions by preventing the establishment of unauthorized sessions and by reducing the chances for session reset attacks. rACLs however, are not effective mitigating attacks sourced from trusted routers and based on trusted protocols. It should also be noted that rACLs apply to traffic destined to the RP, and does not affect transit traffic. rACLs are explained in more detail in Chapter 4, "Device Resiliency and Survivability."

Control Plane Policing and Protection

Control Plane Policing (CoPP) and Control Plane Protection are security infrastructure features that allow the configuration of QoS policies that rate limit the traffic sent to the RP in Cisco IOS software-based devices. Control Plane Policing (CoPP) works best on modular platforms with distributed processing power such as Cisco 12000 Series Routers and the Catalyst 6500 Series Switches, while Control Plane Protection is best suited for non-distributed, software-only platforms such as Cisco ISR Routers. Both features help protect routers from unauthorized access and DoS attacks, even when they originate from valid sources and for valid protocols. Both features also help protect routing sessions by preventing the establishment of unauthorized sessions, and by reducing the chances for session reset attacks. These features are explained in more detail in Chapter 4, "Device Resiliency and Survivability."

Route Filtering

Route filtering is another import tool to secure the routing infrastructure. Most routing protocols allow the configuration of route filters that prevent specific routes from being propagated throughout the network. In terms of security, these filters are useful because they help ensure that only legitimate networks are advertised; and that networks that are not supposed to be propagated are never advertised, i.e. networks falling within the private address space (RFC 1918) should not be advertised out to the Internet.

Route filtering can be divided in two forms, filtering of routing information exchanged between routing peers, and filtering of the routing information exchanged between routing processes in the same router as a result of redistribution. Both forms of route filtering are covered in this document

Cisco IOS provides a series of features used to control the propagation of routing data:

- Route Maps
- Prefix List
- Distribute Lists
- Peer Prefix Filtering
- Maximum Prefix Filtering
- EIGRP Stub Routing
- Route Redistribution Filtering

Route Maps

Similarly to access control lists (ACLs), route maps are used for a variety of purposes including, but not limited to, packet classification, policy routing, address translation, and route filtering. Like ACLs, route maps allow the definition of criteria matching packets or routes along with enforceable actions. But unlike ACLs that only enforce permit and deny actions, route maps allow the reconfiguration of a wide range of parameters in packets and routes, such as next-hop, community, metric, etc. Overall, route maps provide greater policy flexibility and granularity than ACLs. Since the interest of this document is on route filtering, this section will focus on the use of route maps for peer route and route distribution filtering.

Route maps are configured with a tag-name used for identification. Route maps may consist of several entries organized in sequence. Each route map entry contains a list of **match** and **set** statements. The **match** statements specify the match criteria, which are the conditions under which a route will be matched for the given entry. The **set** statements specify the set actions, which are the particular filtering actions to be performed if the criteria specified by the **match** statements are met. Similar to ACLs, route map entries are processed sequentially until a match occurs. When a match occurs the **set** actions for that entry are applied. Any route that does not match at least one match clause relating to a route-map command is simply ignored; that is, the route will not be advertised or redistributed.

The following example shows a two-entry route-map configured to control the redistribution of routes from OSPF into EIGRP. As the route-map is inspected sequentially, external type-2 routes will match the first entry. Since this entry has a **deny** action associated it will prevent those routes from being injected into EIGRP. OSPF routes other than external type-2 will match the second entry of the route-map with a **permit** statement, allowing the redistribution of these routes into EIGRP. In addition, routes matching the second statement will be reconfigured with the metric parameters specified in the **set** statement.

```
route-map ospf-to-eigrp deny 10
 match route-type external type-2
route-map ospf-to-eigrp permit 20
 set metric 40000 1000 255 1 1500
!
router eigrp 1
 redistribute ospf 1 route-map ospf-to-eigrp
 default-metric 20000 2000 255 1 1500
```

As shown in the previous example, route maps are used in conjunction with the **redistribute** command to control the redistribution of routing information between routing processes. At the same time, route maps may be used with the **distribute-list** command to filter routing updates between peers. Both uses are explained with detail later in this document.

Prefix List

An **ip prefix-list** is list of prefixes used in BGP to control the propagation of inbound and outbound updates between routing peers. Similar to ACLs, each entry in the prefix-list contains an IP address and a bit mask, and it is associated with a **permit** or **deny** keyword to either permit or deny the prefix based on the matching condition. The IP address defined in each prefix-list entry can be a classful network, a subnet, or a single host route; while the bit mask is entered as a number from 1 to 32. Similarly to ACLs, prefix-lists are processed sequentially until a match is made; likewise an implicit **deny** is applied to routes that do match any prefix-list entry. Prefix lists can be configured to match an exact prefix length or a prefix range. Prefix ranges can be defined by using the **ge** (greater than or equal) and the **le** (less than or equal) keywords.

A prefix list may be applied to filter inbound or outbound updates for a specific peer or for all peers. Updates from or to specific peers can be controlled by using an **ip prefix-list** in conjunction with the **neighbor prefix-list** command. Updates to or from all peers can be filtered by using an **ip prefix-list** in conjunction with the **distribute-list** command. Both cases are covered in detail later in this document.

The following example applies the prefix list named CustomerA to outgoing advertisements to neighbor 198.133.219.10. The prefix-list CustomerA only allows the advertisement of network 192.133.0.0 to that neighbor:

```
ip prefix-list CustomerA permit 64.104.0.0/16
ip prefix-list CustomerA deny 0.0.0.0/0 le 32
!
router bgp 10
  network 64.104.0.0
  neighbor 198.133.219.10 prefix-list CustomerA out
```

In the following example, a prefix list and distribute list are defined to configure the BGP routing process to advertise only network 64.104.0.0 to all peers.

```
ip prefix-list AllCustomers permit 64.104.0.0/16
ip prefix-list AllCustomers deny 0.0.0.0/0 le 32
!
router bgp 10
  distribute-list prefix AllCustomers out
!
```

Distribute List

A **distribute-list** is a command configured within a routing process that controls what routes are accepted or advertised based on a criteria set by a standard ACL. A **distribute-list** affects all routes being received or advertised by the routing process, including updates between peers as well as routes derived from redistribution. Distribute-lists are unidirectional; the **distribute-list out** command filters outgoing routing updates, while the **distribute-list in** command controls routes received in updates.

In BGP, a **distribute-list** can be associated with not only a standard ACL but also with extended ACL or a **prefix-list**. In addition, BGP distribute-lists can be defined per neighbor with the **neighbor distribute-list** command. Neighbor distribute-lists allow the definition of per-neighbor route policies.

The following example illustrates the use of a distribute-list in EIGRP to control reception of routing updates from all neighbors. In this example, EIGRP process 100 is configured to accept two networks only from any routing neighbor, networks 0.0.0.0 and 10.0.0.0.

```
access-list 1 permit 0.0.0.0
access-list 1 permit 10.0.0.0
!
router eigrp 100
 network 10.0.0.0
 distribute-list 1 in
```

This example illustrates the use of a distribute-list to control the redistribution of routes. In this example, access list 11 allows OSPF to redistribute information learned from RIP only for network 10.139.0.0:

```
router ospf 100
 redistribute rip subnet
 distribute-list 11 out rip
!
access-list 11 permit 10.139.0.0 0.0.255.255
```

Peer Prefix Filtering

Prefix filtering is a practice that helps prevent the injection and propagation of invalid route information from routing peers. False routing information may be injected intentionally or by mistake causing unnecessary traffic redirection, and even triggering DoS conditions on the network infrastructure. Prefix filtering consists in the enforcement of route filters that block the advertisement and reception of unwanted routing updates. This practice also helps reduce the amounts of resources required for generating and processing routing updates.

Prefix filtering is deployed at the edge of routing domains, for instance in an enterprise at the internet edge; and it provides better results when applied to both reception and propagation of routing updates (inbound and outbound updates). In both cases, filtering may be designed to either block unwanted route updates explicitly, or to permit only known valid route prefixes. For example, it is a good practice for SPs to deploy ingress filters at the customer edge to only accept the prefixes assigned or allocated to downstream customers. At the same time, SPs should deploy ingress filters at the peering links to explicitly block common known bogus address spaces and special-use IPv4 addresses.

IGP Prefix Filtering

In Interior Gateway Protocols (IGPs) like EIGRP, IGRP, RIP and OSPF, the exchange of routing information between peers can be controlled with distribute lists. Distribute lists are defined within the routing process and are unidirectional, therefore for a given process two distribute lists can be defined, one for inbound routing updates, and another one for outbound routing updates. This allows for greater policy granularity. In these IGPs, the criteria used by distribute lists to block or permit specific routes is defined via standard ACLs.

Inbound routing updates are filtered by using the **distribute-list in** router configuration mode command:

```
Router(config-router)# distribute-list [[access-list-number | name] | [route-map map-tag]]
in [interface-type | interface-number]
```

- *access list numbe*r or *name*—Number of name of the standard IP access list that defines which networks are to be received and which are to be suppressed in routing updates.
- *map tag*—(OSPF Only) Name of the route map that defines which networks are to be installed in the routing table and which are to be filtered from the routing table. This argument is supported by OSPF only.
- *in*—Applies the access list to inbound routing updates.
- *interface type and number*—Interface type and number on which the access list should be applied to inbound updates. If no interface is specified, the access list will be applied to all inbound updates. The interface type and number arguments can apply if you specify an access list, not a route map.

For a given routing process, it is possible to define one inbound interface-specific distribute-list per interface, and one globally-defined distribute-list. For example, the following combination is possible:

```
access-list 1 permit 10.0.0.0 0.255.255.255
access-list 2 permit 10.122.139.0 0.0.0.255
router rip
 distribute-list 2 in ethernet 0
 distribute-list 1 in
```

In the example above, the router checks the interface on which the update comes in. If it is Ethernet 0, access-list 2 is applied before putting it in the routing table. If, based on this check, the network is denied, no further checking is done for this network. However, if distribute-list 2 allows the network, then the globally-defined distribute-list 1 is also checked. If both distribute-lists allow the network, it is put in the table.

Outbound routing updates are filtered by using the **distribute-list out** router configuration mode command:

```
distribute-list {access-list-number | access-list-name} out [interface-name |
routing-process | as-number]
```

- *access list number or name*—The list defines which networks are to be sent and which are to be suppressed in routing updates.

- *interface name*— Interface type and number on which the access list should be applied to outbound updates. If no interface is specified, the access list will be applied to all outgoing updates.
- *routing process and autonomous system number*—This applies to redistribution. It indicates the routing process and AS number to which routes are distributed. Route redistribution is controlled by the access list.

In the following example, the router only sends routes pertaining to the 10.122.139.0 subnet out of Ethernet 0, and any updates about networks in the 10.0.0.0 are flooded out to the remaining interfaces, including the 10.122.139.0 subnet.

```
access-list 1 permit 10.0.0.0 0.255.255.255
access-list 2 permit 10.122.139.0 0.0.0.255
router rip
 distribute-list 2 out ethernet 0
 distribute-list 1 out
```

BGP Prefix Filtering

Similarly to IGPs, distribute lists can be used in BGP to filter the exchange of routing information with all peers. Distribute lists are defined within the BGP routing process and are unidirectional, therefore for a given process two distribute lists can be defined, one for inbound routing updates, and another one for outbound routing updates. Unlike IGPs, in addition to standard ACLs, distribute lists in BGP support extended ACLs and prefix-lists.

Inbound routing updates may be filtered by using the **distribute-list in** router configuration mode command:

```
distribute-list {acl-number | prefix list-name} in
```

- *access list number*—Standard or extended IP access list that defines which networks are to be received and which are to be suppressed in routing updates.
- *prefix list-name*— The prefix list defines which networks are to be received and which are to be suppressed in routing updates, based upon matching prefixes. IP prefix lists are used to filter based on the bit length of the prefix. An entire network, subnet, supernet, or single host route can be specified.

In the following example, a prefix list and distribute list are defined to configure the BGP routing process to accept traffic from only network 198.133.219.0 and network 64.104.0.0.

```
ip prefix-list RED deny 0.0.0.0/0 le 32
ip prefix-1ist RED permit 64.104.0.0/16
ip prefix-list RED permit 198.133.219.0/24
router bgp 10
  network 64.104.0.0
  distribute-list prefix RED in
```

Outbound routing updates may be filtered by using the **distribute-list out** router configuration mode command:

```
distribute-list {acl-number | prefix list-name} out [protocol process-number | connected |
static]
```

- *access list number*—Standard or extended IP access defining which networks are to be received and which are to be suppressed in routing updates.
- *prefix list-name*—The prefix list defines which networks are to be received and which are to be suppressed in routing updates, based upon matching prefixes in the prefix list.

- **protocol** *process-number*—This applies in case of redistribution. It specifies the routing protocol to apply the distribution list. BGP, EIGRP, OSPF, and RIP are supported. The process number is entered for all routing protocols, except RIP. The process number is a value from 1 to 65535.
- **connected**—Specifies peers and networks learned through connected routes.
- **static**—Specifies peers and networks learned through static routes.

In the following example, a prefix list and distribute list are defined to configure the BGP routing process to advertise only network 192.133.219.0.

```
ip prefix-list BLUE deny 0.0.0.0/0 le 32
ip prefix-list BLUE permit 192.133.219.0/24
router bgp 10
  distribute-list prefix BLUE out
```

BGP routing updates can also be controlled on a per neighbor basis. BGP provides four mechanisms to do that: AS-path filters (not covered in this document), **neighbor distribute-list**, **neighbor prefix-list** and **neighbor route-map**.

The **neighbor distribute-list** command allows you to control inbound and outbound routing updates on a per neighbor basis. The filtering criteria can be set by using a standard, an extended ACL, or a prefix list. Standard access lists may be used to filter routing updates. However, in the case of route filtering when using classless interdomain routing (CIDR), standard access lists do not provide the level of granularity that is necessary to configure advanced filtering of network addresses and masks, so extended access lists should be used instead. Neighbor distribute-lists are unidirectional, therefore for a given neighbor only two distribute lists can be applied, one per direction. Another characteristic of neighbor distribute-lists is that they can be applied to individual neighbors or on peer groups. When a peer group-name is used, all the members of the peer group will inherit the characteristic configured with this command. Specifying the command for a neighbor overrides the inbound policy that is inherited from the peer group.

```
neighbor {ip-address | peer-group-name} distribute-list {access-list-number |
expanded-list-number | access-list-name| prefix-list-name} {in | out}
```

- *ip-address*—IP address of the neighbor.
- *peer-group-name*—Name of a BGP peer group.
- *access list number* or *name*—Number of a standard, number or name of the extended access list that defines which networks are to be received and which are to be suppressed in routing updates.
- *prefix list name*—Name of a BGP prefix list.
- *in*—Access list is applied to inbound advertisements to that neighbor.
- *out*—Access list is applied to outbound advertisements to that neighbor.

The following router configuration mode example applies list 39 to inbound advertisements from neighbor 198.133.219.10. List 39 permits the advertisement of network 64.104.0.0.

```
access-list 39 permit 64.104.0.0 0.0.255.255
!
router bgp 10
 network 64.104.0.0
 neighbor 198.133.219.10 distribute-list 39 in
```

The **neighbor prefix-list** command allows you to control inbound and outbound routing updates on a per neighbor basis based on the bit length of the prefixes. An entire network, subnet, supernet, or single host route can be specified. Neighbor prefix-lists are unidirectional, therefore for a given neighbor only two distribute lists can be applied, one per direction. Neighbor prefix-lists can also be applied to individual neighbors or on peer groups. When a peer group-name is used, all the members of the peer group will inherit the characteristic configured with this command. Specifying the command for a neighbor overrides the inbound policy that is inherited from the peer group.

neighbor {ip-address | peer-group-name} **prefix-list** prefix-list-name {**in** | **out**}

- *ip-address*—IP address of neighbor.
- *peer-group-name*—Name of a BGP peer group.
- *prefix-list-name*—Name of prefix list defining which networks are to be received and which are to be suppressed in routing updates.
- *in*—Filter list is applied to inbound advertisements from that neighbor.
- *out*—Filter list is applied to outbound advertisements to that neighbor.

In the following example, an SP deploys a customer edge prefix filter to only allow the prefix assigned to one customer (64.104.0.0/16).

```
router bgp 101
     neighbor 198.133.219.6  remote-as 10
     neighbor 198.133.219.6 prefix-list customer in
!
ip prefix-list customer permit 64.104.0.0/16
ip prefix-list customer deny 0.0.0.0/0 le 32
```

In BGP it is also possible to control the exchange of routing updates to or from particular neighbors by using route maps. Route maps are a powerful tool that can also be used to modify BGP attributes such as next-hop, communities, local-preference, etc; though such application is beyond the scope of this document.

To control routing updates on a per neighbor basis with route maps, use the **neighbor route-map** command. As the other neighbor filtering mechanisms, neighbor route-maps can be applied in each direction, one to control the reception of routing updates, and another one to filter the advertisement of routing updates. Neighbor route-maps can also be applied to individual neighbors or on peer groups. When a peer group-name is used, all the members of the peer group will inherit the characteristic configured with this command. Specifying the command for a neighbor overrides the inbound policy that is inherited from the peer group.

neighbor {ip-address | peer-group-name } **route-map** map-name {**in** | **out**}

- *ip-address*—IP address of the neighbor.
- *peer-group-name*—Name of a BGP or multiprotocol BGP peer group.
- *map-name*—Name of a route map that defines which networks are to be received and which are to be suppressed in routing updates. When a route map is specified, only routes that match at least one section of the route map are received or advertised.
- *in*—Applies route map to inbound routes.
- *out*—Applies route map to outbound routes.

In the following, the **route-map localonly** command allows only the locally generated routes to be advertised to neighbor 198.133.219.10. This prevents the risk of the autonomous system becoming a transit AS for Internet traffic.

```
ip as-path access-list 10 permit ^$
route-map localonly permit 10
  match as-path 10
!
router bgp 10
 network 64.104.0.0
 neighbor 198.133.219.10 remote-as 100
 neighbor 198.133.219.10 route-map localonly out
```

See the *Cisco IOS IP Routing Protocols Configuration Guide* at the following URL for your IOS Software Release to learn how to configure prefix filtering with your routing protocol.

http://www.cisco.com/web/psa/products/tsd_products_support_configure.html

Maximum Prefix Filtering

Some routing protocols allow the definition of the maximum number of routes to be accepted from a routing peer. When this feature is enabled on a router and once the limit has been exceeded, by default the router tears down the peering session, clears all routes learnt from the peer, and then places the peer in a penalty state for a time period. After the penalty time period expires, normal peering is reestablished. This capability helps protect the router from attacks based on the injection of large volumes of routes and unintentional configuration mistakes leading to Denial of Service conditions. Setting a Maximum Prefix limit is particularly useful on routers at the border of routing domains.

In BGP and EIGRP, maximum prefix filtering is configured with the **neighbor maximum-prefix** command. OSPF has a similar feature that limits the number of nonself-generated link-state advertisements an OSPF routing process can keep in the OSPF link-state database. In OSPF, this feature is configured with the **max-lsa** command.

While most routing protocols by default respond by tearing down a session when the limit is exceeded, it is highly recommended to initially configure the routers to alarm only. In that case the router, instead of tearing down the session, only sends a log message, and it continues peering with the sender.

To illustrate the concept, the following EIGRP configures the maximum prefix limit for a single peer. The maximum limit is set to 1000 prefixes, and the warning threshold is set to 80 percent. When the maximum prefix limit is exceeded, the session with this peer will be torn down, all routes learned from this peer will be removed from the topology and routing tables, and this peer will be placed in a penalty state for 5 minutes (default penalty value).

```
Router(config)# router eigrp 100
Router(config-router)# neighbor 10.0.0.1 maximum-prefix 1000 80
```

See the *Cisco IOS IP Routing Protocols Configuration Guide* at the following URL for your IOS Software Release to learn how to configure maximum prefix filtering with your routing protocol.

http://www.cisco.com/web/psa/products/tsd_products_support_configure.html

EIGRP Stub Routing

EIGRP allows the configuration of stub routers in hub and spoke topologies where the spoke routers direct all traffic to a hub distribution router. The stub configuration helps control the propagation of routing information from the spoke router, preventing both the distribution of false routing data and the manipulation of valid routes. The stub router can be configured to permit the propagation of static, connected, and summary routes independently. In the case a bogus router is installed in the stub network, and assuming the bogus router successfully establishes a peering session, the stub router will learn any routing data originated from the bogus router but it will not pass it onto other peers, shielding the rest of the routing infrastructure from the potential injection of faulty routing data.

Configuring EIGRP Stub Routing is a recommendable practice for stub routers that are part of a hub and spoke topology. In practice, EIGRP Stub Routing provides a layer of control on route propagation that helps reinforce the route filtering techniques described earlier in this document. EIGRP Stub Routing is complementary to those techniques and not a replacement.

Note Even though OSPF implements the concept of stub routing, its implementation differs from EIGRP Stub Routing. Configuring an OSPF area as stub does not prevent a router connected to the stub area from accidentally or intentionally injecting invalid routing information.

Stub routing can be configured in EIGRP with the **eigrp stub** command. In the following example, the **eigrp stub** command is issued with the **connected** and **static** keywords to configure the router as a stub that advertises connected and static routes (sending summary routes will not be permitted):

```
router eigrp 100
 network 10.0.0.0
 eigrp stub connected static
```

See the *Cisco IOS IP Routing Protocols Configuration Guide* at the following URL of your IOS Software Release for more information on stub routing.

http://www.cisco.com/web/psa/products/tsd_products_support_configure.html

Route Redistribution Filtering

In networks where route redistribution is required, it is a good practice to strictly control which routes are advertised as well as to limit the maximum number of routes learned from another routing domain. Invalid routes intentionally or unintentionally introduced into a routing domain may be propagated throughout redistribution leading to the bypass of security controls and even creating denial of service conditions. Implementing route redistribution filters helps contain the effects of such conditions. In addition, from a pure routing standpoint, redistribution filters may help prevent a routing process from sending routing updates back to the process from where they have been originated, preventing loops and maintaining network stability.

Route redistribution is configured with the **redistribute** command. Filters can be enforced by either associating a **route-map** to the redistribute command, or by using a **distribute list** in conjunction with the redistribute command.

To illustrate, the following is the syntax of the redistribute command. Note that the redistribute command support more parameters than the ones shown here and that were omitted for simplicity.

```
redistribute protocol [process-id][as-number][route-map map-tag]
```

- *Protocol*—Source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **connected**, **eigrp**, **isis**, **mobile**, **ospf**, **static** [ip], or **rip**.
- Process ID—For the **bgp** or **eigrp** keyword, this is an autonomous system number, which is a 16-bit decimal number.
- *AS-number*: Autonomous system number for the redistributed route.
- *Route-map and map-tag*—Route map that should be interrogated to filter the importation of routes from this source routing protocol to the current routing protocol. If not specified, all routes are redistributed. If this keyword is specified, but no route map tags are listed, no routes will be imported.

When used with the **redistribute** command, a **route-map** serves as an inbound filter, controlling only the importation of routes from the source routing protocol into the current protocol. To control the redistribution of routing updates in both directions either configure a **route-map** in each routing process, or define an outbound filter in the current routing protocol with the **distribute-list out** command.

The following example illustrates the use of **route-map** with the **redistribute** command. In this example, routes are being redistribute between EIGRP and RIP. Route map **rip**-to-**eigrp** prevents the import of network 10.0.0.0/8 into EIGRP. Likewise, route map **eigrp**-to-**rip** prevents the import of network 20.0.0.0/8 into RIP.

```
route-map rip-to-eigrp deny 10
 match ip address 1
route-map rip-to-eigrp permit 20
!
route-map eigrp-to-rip deny 10
 match ip address 2
route-map eigrp-to-rip permit 20
!
router eigrp 100
 network 10.0.0.0
 redistribute rip route-map rip-to-eigrp
!
router rip
 network 20.0.0.0
 redistribute eigrp 1 route-map eigrp-to-rip
!
access-list 1 permit 10.0.0.0 0.255.255.255
access-list 2 permit 20.0.0.0 0.255.255.255
```

Route redistribution can also be filtered by using **distribute lists** in the same way as explained forPeer Prefix Filtering. Route updates can be filtered simultaneously in both directions by configuring a **distribute-list in** and a **distribute-list out** in the same routing process. The **distribute-list in** controls the import of network routes from all sources into the current routing process, while the **distribute-list out** filters the export of network routes from the current routing process. A **distribute-list out** can be applied to all route advertisements or to those of a particular routing process.

As discussed in Peer Prefix Filtering, page 3-10, in interior gateway protocols the criteria can only be defined with a standard ACL. OSPF however supports route-maps in addition to standard ACLs. BGP distribute lists support extended ACLs and prefix-lists in addition to standard ACLs.

This example shows the same redistribution scenario as the previous example but this time using distribute lists. In this case, the EIGRP process is configured with a distribute list preventing the redistribution of network 10.0.0.0/8 into RIP. Likewise, RIP is configured with a distribute-list that prevents the advertisement of network 20.0.0.0/8 into EIGRP.

```
router eigrp 100
 network 10.0.0.0
 redistribute rip
 distribute-list 10 out rip
!
router rip
 network 20.0.0.0
 redistribute eigrp 1
 distribute-list 20 out eigrp 1
!
access-list 10 deny 10.0.0.0 0.255.255.255
access-list 10 permit 0.0.0.0 255.255.255.255
access-list 20 deny 20.0.0.0 0.255.255.255
access-list 20 permit 0.0.0.0 255.255.255.255
```

Another good practice is to limit the number of prefixes redistributed into a routing process. In OSPF and EIGRP a maximum limit of prefixes can be set by using the **redistribute maximum-prefix** command. When the redistributed maximum-prefix command is configured, and if the number of redistributed routes reaches the maximum value configured, no more routes are redistributed (unless the router is configured to generate a warning message).

In the following OSPF example, a maximum number of 1200 prefixes can be redistributed into OSPF process 1. If the number of prefixes redistributed reaches 80 percent of 1200 (960 prefixes), a warning message is logged. Another warning is logged if the limit is reached, and no more routes are redistributed.

```
router ospf 1
 network 10.0.0.0 0.0.0.255 area 0
 redistribute eigrp 10 subnets
 redistribute maximum-prefix 1200 80
```

See the *Cisco IOS IP Routing Protocols Configuration Guide* at the following URL to learn how to configure route redistribution filtering with your routing protocol.

http://www.cisco.com/web/psa/products/tsd_products_support_configure.html

Logging

Frequent neighbor status changes (up or down) and resets are common symptoms of network connectivity and network stability problems that should be investigated. These symptoms may also indicate ongoing attacks against the routing infrastructure. Logging the status changes of neighbor sessions is a good practice that helps identify such problems and that facilitates troubleshooting. In most routing protocols, status change message logging is enabled by default. When enabled, every time a router session goes down, up, or experiences a reset, the router generates a log message. If syslog is enabled, the message is forwarded to the syslog server; otherwise is kept in the router’s internal buffer.

To enable status change message logging in BGP use the **bgp log-neighbor-changes** command, in EIGRP use the **eigrp log-neighbor-changes** command, and in OSPF use the **log-adjacency-changes** command.

The following example logs neighbor changes for BGP in router configuration mode:

```
Router(config)# router bgp 10
Router(config-router)# bgp log-neighbor-changes
```

See the *Cisco IOS IP Routing Protocols Configuration Guide* at the following URL to learn how to configure route redistribution filtering with your routing protocol.

http://www.cisco.com/web/psa/products/tsd_products_support_configure.html

Secure Routing Plane Summary

Table 3-3 summarizes the best practices explained in this document and lists the threat they help mitigate.

Table 3-3 Secure Routing Plane Best Practices Summary

Restricted Routing Protocol Membership	**Router**	**Peering Session**	**Routing Data**
Neighbor authentication		session resets, unauthorized peering	injection of false routes, removal or modification of routes
Routing peer definition		unauthorized peering	interception of routing information
Default passive interface	Denial-of-Service (DoS)	unauthorized peering	
BGP TTL Security Check	DoS	unauthorized peering, session resets	
iACLs	unauthorized access, invalid source/protocol DoS	session resets, unauthorized peering	
rACLs	unauthorized access, invalid source/protocol DoS	session resets, unauthorized peering	
Control Plane Policing	unauthorized access, invalid/valid DoS, saturate router queues	session resets, unauthorized peering	
Control Plane Protection	unauthorized access, invalid/valid DoS, saturate router queues	session resets, unauthorized peering	
Route Filtering	**Router**	**Peering Session**	**Routing Data**
Peer Route Filtering	DoS-based on routing info		injection of false routes
Max Prefix Filtering	DoS-based on routing info		injection of false routes
EIGRP Stub Routing	DoS-based on routing info		injection of false routes, removal or modification of routes
Route Redistribution	DoS-based on routing info		injection of false routes, removal or modification of routes

CHAPTER 4

Device Resiliency and Survivability

Routers and switches may be subject to attacks designed to or that indirectly affect the network availability. Possible attacks include DoS based on unauthorized and authorized protocols, Distributed DoS, flood attacks, reconnaissance, unauthorized access and more. This section presents a collection of best practices destined to preserve the resiliency and survivability of routers and switches, helping the network maintain availability even during the execution of an attack.

CSF Methodology Assessment

The results of applying the CSF methodology are presented in table x and highlight the technologies and features identified for ensuring device resiliency and survivability, and which are integrated in Network Secure Baseline.

Total Visibility

Table 4-1 Device Resiliency and Survivability—Total Visibility

Identify	Monitor	Correlate
Port Security	• Logging – Syslog – SNMP	

Complete Control

Table 2 Device Resiliency and Survivability—Complete Control

Harden	Isolate	Enforce
• Control Plane Policing • Control Plane Protection • Backup interfaces • Element redundancy – HAS – RPR, RPR+ – SSO • Standby devices – Failover – HSRP – VRRP – GLBP • Topological Redundancy		• Disable Unnecessary Services • iACLs

Disabling Unnecessary Services

To facilitate deployment, Cisco routers and switches come out of the box with a list of services turned on that are considered appropriate for most network environments. However, since not all networks have the same requirements, some of these services may not be needed and in consequence can be disabled. Disabling these unnecessary services has two benefits. It helps preserve system resources, and eliminates the potential of security exploits on the disabled services.

This section describes how to disable some services that may not be needed. As an alternative, Cisco IOS software provides the `AutoSecure` CLI command that helps disable these unnecessary services, while enabling other security services.

Note Before disabling a service make sure the service is not needed.

This section describes the procedure for disabling the following services, which are typically not needed in infrastructure networks:

- Cisco Discovery Protocol (CDP)
- Directed Broadcast
- Finger
- Maintenance Operations Protocol (MOP)
- IP BOOTP Server
- IP Redirects
- IP Source Routing

- PAD
- Proxy ARP
- Ident
- TCP and UDP Small Servers

Cisco Discovery Protocol (CDP)

Cisco Discovery Protocol (CDP) is a Cisco proprietary Layer 2 protocol designed to facilitate the administration and troubleshooting of network devices by providing information on neighboring equipment. With CDP enabled, network administrators can execute CDP commands that provide them with the platform, model, software version, and even the IP addresses of adjacent equipment.

CDP is a useful protocol, but potentially could reveal important information to an attacker. CDP is enabled by default, and can be disabled globally or for each interface. The best practice is to disable CDP globally when the service is not used, or per interface when CDP is still required. In cases where CDP is used for troubleshooting or security operations, CDP should be left enabled globally, and should be disabled only on those interfaces on which the service may represent a risk, for example, interfaces connecting to the Internet. As a general practice, CDP should not be enabled on interfaces that connect to external networks, such as the Internet.

To disable CDP globally use the no cdp run command from global configuration mode, as in the following example:

```
Router(config)# no cdp run
```

To disable CDP on one or more interfaces, use the no cdp enable command from interface configuration mode, as in the following example:

```
Router(config-if)# no cdp enable
```

Note Features such as ODR (on demand routing) depend on CDP, so check for dependencies prior to disabling CDP.

For more information about CDP, refer to the following URL:

http://www.cisco.com/en/US/partner/tech/tk962/technologies_tech_note09186a00801aa000.shtml

Directed Broadcast

An IP directed broadcast packet is an IP packet whose destination address is a valid broadcast address for an IP subnet. When a directed broadcast packet reaches a router that is directly connected to its destination subnet, and if the router is configured to do so, that packet is “exploded” as a broadcast on the destination subnet. By default, earlier releases of Cisco IOS software handle directed broadcasts this way. However, because directed broadcasts have been used for attacks, such as the SMURF attack, the default behavior has been changed to drop directed broadcasts since Cisco IOS software Release 11.2.

In the case the forwarding of directed broadcast has been enabled, or in the case of Cisco IOS software releases prior to Cisco IOS software Release 11.2, it is s recommended that you disable this feature on all interfaces using the **no ip directed-broadcast** interface configuration command, as in the following example:

```
Router(config-if)# no ip directed-broadcast
```

For more information about the ip directed-broadcast command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/ipaddr/command/reference/ip1_i1g.html#wp1081245

Finger

Finger, as defined in RFC 742, is a protocol that can be used to obtain information about users logged into a remote host or network device. Cisco IOS software incorporates a finger service, which in Cisco IOS software releases prior to 12.1(5) and 12.1(5)T was turned on by default. Although the finger service does not reveal any extremely sensitive information, it can be used by a potential attacker to gather information. Therefore it is recommended that you disable this service.

In older releases of Cisco IOS software where the finger service was enabled by default, it can be disabled with the **no service finger** global configuration command, as in the following example:

```
Router(config)# no service finger
```

Starting in Cisco IOS software 12.1(5) and 12.1(5)T, the finger service is disabled by default. If finger has been turned on and the service is not needed, it can be disabled with the **no ip finger** global configuration command, as in the following example:

```
Router(config)# no ip finger
```

For more information on the finger service, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g03.html#wp1033299

Maintenance Operations Protocol (MOP)

The Maintenance Operations Protocol (MOP) was developed by Digital Equipment Corporation to be used for remote communications between hosts and servers. Cisco IOS software routers implement MOP to gather configuration information when communicating with DECNet networks. By default, MOP is enabled on all Ethernet interfaces, and disabled on all other type of interfaces. The MOP service can be disabled per interface by using the **no mop enabled** interface configuration command, as in the following example:

```
Router(config-if)# no mop enabled
```

MOP has been proven vulnerable to various attacks; therefore it should be disabled on all access and externally facing interfaces unless they provide connectivity to DECNet networks.

For more information about the **mop enabled** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/interface/command/reference/int_m1g.html#wp1139514

IP BOOTP Server

As defined by RFC 951, the Bootstrap protocol allows a diskless workstation to configure itself at boot time by dynamically obtaining an IP address, the IP address of the BOOTP server, and a configuration file. Cisco IOS software implements a bootstrap service that allows a router to act as a BOOTP server providing dynamic configuration services to other Cisco IOS software routers. This service is turned on by default and it is used by features like AutoInstall, which simplifies or automates the configuration of Cisco devices. If not needed, this service should be disabled with the **no ip bootp server** global configuration command, as in the following example:

```
Router(config)# no ip bootp server
```

For more information about the BOOTP server service, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g03.html#wp1031545

For more information about AutoInstall, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf002.html

IP Redirects

By default, Cisco IOS software sends ICMP redirect messages when it is forced to resend a packet through the same interface on which it was received. By sending these redirect messages the router instructs the host the specific router to use to reach a particular destination. The ICMP redirect messages can also reveal information that can potentially be used by an attacker for discovering the network topology. Therefore, it is highly recommend that you disable this service on all access and externally facing interfaces. IP redirects can be disabled on each interface using the **no ip redirects** interface configuration command, as in the following example:

```
Router(config-if)# no ip redirects
```

For more information about the **ip redirect** command, see the following website:

http://www.cisco.com/en/US/docs/ios/12_3/ipaddr/command/reference/ip1_i2g.html#wp1081518

Note Previously, if the Hot Standby Router Protocol (HSRP) was configured on an interface, ICMP redirect messages were disabled by default for the interface. With Cisco IOS software Release 12.1(3)T, ICMP redirect messages are enabled by default even if HSRP is configured.

IP Source Routing

The IP protocol supports source routing options that allow the sender of an IP packet to control the route that the datagram will take toward its ultimate destination, and generally the route that any reply will take. These options are rarely used for legitimate purposes in real networks. Some older IP implementations do not process source-routed packets properly, and it may be possible to crash machines running these implementations by sending them datagrams with source routing options. By default Cisco IOS software forwards IP packets with source routing header options. As a general best practice, IP source routing should be disabled unless strictly necessary. To have the software discard any IP packet containing a source-route option, use the **no ip source-route** global configuration command as in the following example:

```
Router(config)# no ip source-route
```

For more information about the **ip source-route** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/ipaddr/command/reference/ip1_i2g.html#wp1081830

PAD

Cisco IOS software provides a PAD (packet assembler/disassembler) service that allows simple devices such as character-mode terminals to connect to legacy X.25 networks. With this service Cisco IOS software devices and other X.25 network equipment can establish PAD sessions. By default, the PAD service is enabled on Cisco IOS software, but it could be used to gain unauthorized or inappropriate access. Therefore, unless needed, this service should be disabled with the **no service pad** global configuration command, as in the following example:

```
Router(config)# no service pad
```

For more information about the PAD service, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/wan/command/reference/wan_s1g.html#wp1032441

Proxy ARP

Proxy Address Resolution Protocol (ARP), as defined in RFC 1027, is a technique that helps machines on a subnet reach remote subnets without configuring routing or a default gateway. Proxy ARP is typically implemented on routers, and when configured, the router answers all ARP requests on the local subnet on behalf of systems some hops away.

In this model, local hosts send ARP requests for each destination for which they do not have any routing information, and the router replies with its own MAC address as the next hop. Cisco IOS software by default implements proxy ARP on all interfaces. However, unless strictly needed it should be disabled with the **no ip proxy-arp** interface configuration command, as in the following example:

```
Router(config-if)# no ip proxy-arp
```

For more information about the **ip proxy-arp** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/ipaddr/command/reference/ip1_i2g.html#wp1081466

Ident

As defined by RFC 1413, the TCP Client Identity Protocol (Ident) is a protocol that allows a system to query the identity of a user initiating a TCP connection or a host responding to a TCP connection. When implemented, the Ident service allows a user to obtain identity information by simple connecting to a TCP port on a system, and issuing a simple text string requesting information. This clearly can yield information that could be used to attack the system. Cisco IOS software routers implement an Ident service, which is disabled by default. It is highly recommended that you do not enable this service. If the Ident service has been enabled, it can be disabled by using the **no ip identd** global configuration command, as in the following example:

```
Router(config)# no ip identd
```

For more information about the **ip identd** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/11_2/security/command/reference/2rauthen.html#wp2545

TCP and UDP Small Servers

TCP and UDP small servers are daemons that typically run on Unix systems and that were designed for diagnostic purposes. Cisco IOS software also provides an implementation of UDP and TCP small servers that enables echo, chargen, daytime, and discard services. Unless strictly necessary, these services should be disabled because they can be used by a potential attacker to gather information, or to directly attack the Cisco IOS software device.

TCP and UDP small services are enabled by default on Cisco IOS software Release 11.2 and earlier. These commands are disabled by default on Cisco IOS software Software Versions 11.3 and later.

These commands may be disabled using the **no service tcp-small-servers** and **no service udp-small-servers** global configuration commands, as shown in the following example:

```
Router(config)# no service tcp-small-servers
Router(config)# no service udp-small-servers
```

For more information about TCP and UDP small servers, refer to the following URL:

http://www.cisco.com/en/US/partner/products/sw/iosswrel/ps1818/products_tech_note09186a008019d97a.shtml#tcp_udp_servers

Infrastructure Protection Access Control Lists (iACLs)

Infrastructure protection access control lists (iACLs) is an access control technique that shields the network infrastructure from internal and external attacks. iACLs is not a feature per se, it is a technique based on extended ACLs developed initially by Internet Service Providers (ISPs) to protect their network infrastructures, but that uses concepts that can be leveraged by enterprises as well.

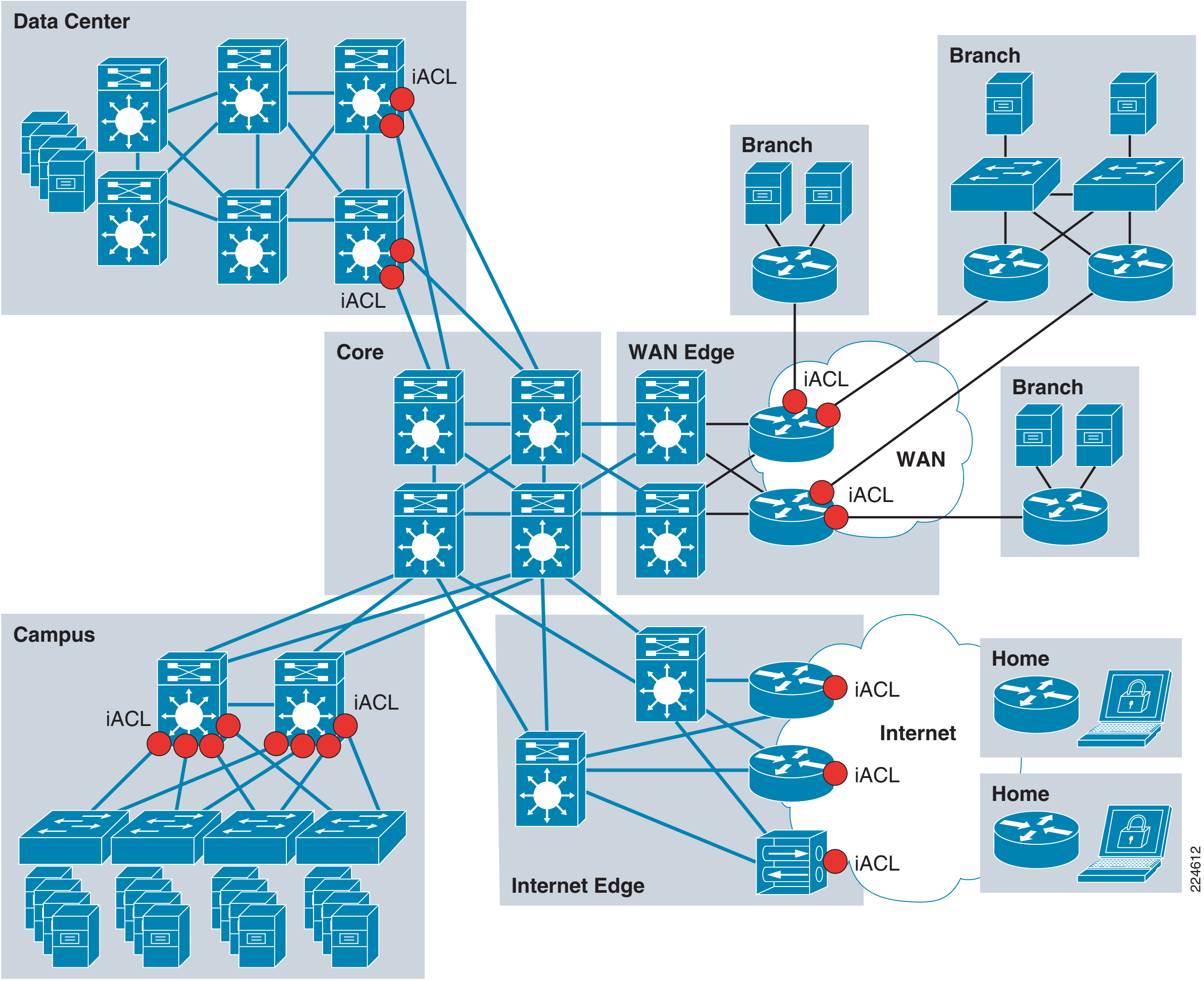
In a nutshell, iACLs are extended ACLs designed to explicitly permit authorized control and management traffic bound to the infrastructure equipment such as routers and switches, while denying any other traffic directed to the infrastructure address space. For example, an iACL deployed at an ISP peering edge is configured to explicitly permit BGP sessions from known peers, while denying any other traffic destined to the ISP’s peering router as well as to the rest of the infrastructure address space.

By only allowing authorized control and management traffic, iACLs help protect routers from unauthorized access and DoS attacks based on unauthorized protocols and sources. In addition, iACLs protect routing sessions by preventing the establishment of unauthorized sessions, and by reducing the chances for session reset attacks. iACLs also help prevent the injection, modification or removal of routing information. It should be noted however, that iACLs are not effective mitigating attacks originated from trusted sources and based on trusted protocols.

iACLs are most useful when deployed at the network edges, where the infrastructure becomes accessible to internal or external users; and at administrative borders, where equipment or links under different administration meet. As mentioned in the previous example, an ISP peering edge is a good place for an iACL because it shields the ISP infrastructure from threats coming from peering links. In an enterprise, iACLs may be deployed at the many network edges. iACLs may be deployed at the WAN edge, protecting the core infrastructure from possible threats coming from remote branch offices and partner locations. iACLs may be configured at the campus distribution, protecting the infrastructure from possible attacks originated from the LANs. Similarly, the filters deployed at the enterprise Internet edge may be designed to function as an iACL to shield the infrastructure from external threats. iACLs are also useful at administrative borders. For example, an enterprise Security Operations Center (SOC) team may decide to implement an iACL to protect its equipment from threats originated somewhere else in the enterprise, and despite the fact that iACLs may be already deployed at the WAN edge, Internet edge, campus or somewhere else. The SOC team may decide to do so simply to maintain control of the protection of its own infrastructure, rather than relying on security elements administered by other administrative teams.

Figure 4-1 shows an example deployment of iACL.

Figure 4-1 iACL Deployment Example



As discussed at the beginning of this document, having an adequate design of the address space facilitates deployment of security measures. This statement cannot be truer than with iACLs. As we will see in this section, the degree of summarization, the level of segmentation between the infrastructure equipment and endpoints, have a direct impact on the number of lines and complexity of the iACL. The more erratic the address space is the more complex the iACL will be and the more lines it will have.

For more information about iACLs, refer to the following URL:

http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a00801a1a55.shtml

iACL Structure

An iACL needs to be built in a structured manner recognizing the fact entries are processed sequentially like other ACLs. Though the specifics on how an iACL should be constructed depend on the particular deployment scenario, an iACL generally consists of four distinct modules, which are described next:

- First module—Anti-spoofing entries, entries that block packets with private addresses, special use, and other source addresses known to be invalid for the given environment,
- Second module—Entries providing explicit permission for traffic from legitimate external sources destined to infrastructure addresses. This includes control management traffic like routing protocols, remote access protocols (i.e., SSH and Telnet), network management traffic (i.e., SNMP), etc.
- Third module—**deny** statements for all other traffic from external sources destined to infrastructure addresses.
- Fourth module—**permit** statements for all other normal backbone traffic en route to non-infrastructure destinations.

The first module is designed to block any obvious illegitimate traffic, such as packets arriving with a source IP address belonging to the internal infrastructure address space, as it is an indication of spoofing. In addition, the first section of the ACL should also prevent packets with special use addresses (RFC-3330). In the case of an ISP peering iACL or enterprise internet edge iACL, the first module may also be configured to block packets arriving from the Internet with private source IP addresses (RFC-1918).

Note RFC 3330 defines special use addresses that might require filtering. RFC 1918 defines reserved address space that cannot be used for valid source addresses on the Internet. RFC 2827 provides ingress filtering guidelines.

The following is an example of useful entries for the first module of an iACL constructed for an ISP peering point or an enterprise Internet edge:

```
! Deny your infrastructure space as a source of external packets
access-list 101 deny ip your_infrastructure_block any
!--- Deny special-use address sources.
!--- See RFC 3330 for additional special-use addresses.
access-list 101 deny ip host 0.0.0.0 any
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
access-list 110 deny ip 192.0.2.0 0.0.0.255 any
access-list 110 deny ip 224.0.0.0 31.255.255.255 any
! Deny RFC1918 space from entering AS
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
access-list 101 deny ip 172.16.0.0 0.0.15.255 any
access-list 101 deny ip 192.168.0.0 0.0.255.255 any
```

The second module should allow legitimate control and management traffic (such as BGP, OSPF, SNMP, or SSH) destined to the infrastructure equipment. This requires a clear understanding of the legitimate traffic bound to the infrastructure. An iACL built without the proper understanding of the protocols and the devices involved may end up blocking critical traffic. Unless you are careful, the iACL has the potential of causing a DoS, instead of preventing it.

The following configuration fragment shows how the second section of the iACL would look for an ISP peering point or enterprise Internet edge, assuming that the only legitimate external traffic were eBGP and OSPF packets from specific peers:

```
! Permit eBGP session
access-list 101 permit tcp host bgp_peer host local_ip eq 179
```

```
access-list 101 permit tcp host bgp_peer eq 179 host local_ip
! Permit OSPF
access-list 101 permit ospf host ospf_neighbor host 224.0.0.5
! Permit DR multicast address, if needed
access-list 101 permit ospf host ospf_neighbor host 224.0.0.6
access-list 101 permit ospf host ospf_neighbor host local_ip
```

The third module of the iACL should deny any other traffic destined to the infrastructure address space, as shown in the following example:

```
! Deny all other access to infrastructure
access-list 101 deny ip any your_infrastructure_block
```

The fourth and final module of the iACL may be configured to either allow or deny all traffic, depending on the scenario. In ISP networks, which are transit networks in nature, this module should be configured to permit any other IP traffic. Likewise, in an enterprise inner iACL this module should also be configured to allow any other traffic.

```
! Permit transit traffic (ISP), enterprise inner iACL
access-list 101 permit ip any any
```

Enterprise public networks are typically the destination for traffic (not transit), and therefore the fourth module of an iACL deployed at an internet edge requires some special consideration. Depending on the existence of a firewall and the security policies in place this module may be configured to either allow or deny all other traffic. If the internet edge incorporates a firewall controlling access to the public enterprise network, then the last module of an iACL at the internet edge router may be configured to allow any other traffic (as in the example above). Per contrary, in networks where there are no firewalls or where the Internet edge router acts as a firewall, this module may be configured to specifically permit the protocols and the IP addresses for the public services, with an implicit “deny any” denying the rest of traffic. For example:

```
! Permit only HTTP service traffic (Enterprise public services)
access-list 101 permit tcp any web_servers eq http
! access-list 101 deny ip any any (implicit)
```

This entry allows only HTTP traffic over TCP to a specific network, while an implicit *deny ip any any* denies the rest of the traffic.

iACL Recommended Deployment Methodology

As previously mentioned, an iACL built without the proper understanding of the protocols and the devices involved may end up being ineffective and may even result in a DoS condition. For this reason, it is vital to gain an adequate level of understanding about the legitimate traffic destined to the infrastructure before deploying an iACL.

In some networks, determining the exact traffic profile needed to build the filters required might be difficult. For this reason, this document recommends a conservative methodology for deploying iACLs., using iterative ACL configurations to help identify and incrementally filter unwanted traffic as you gain a better understanding of the traffic on your network.

To deploy iACLs using this conservative methodology, complete the following steps:

Step 1 Identify protocols used in the network using a discovery ACL.

Start by deploying a discovery or classification ACL, which permits all the commonly used protocols that access infrastructure devices.

The discovery ACL should have a source address of **any** and a destination address that encompasses the entire infrastructure IP address space. Logging can be used to develop a list of source addresses that match the protocol **permit** statements. A last line including permitting **ip any any** is required to enable traffic flow.

The objective of configuring the discovery ACL is to determine the protocols that the specific network uses. Use the `log` keyword for analysis to determine what else might be communicating with the router.

Note Although the **log** keyword provides valuable insight into the details of ACL hits, excessive hits to an ACL entry including this keyword might result in an overwhelming number of log entries and possibly high router CPU usage. Only use the **log** keyword for short periods of time as needed to help classify traffic.

Step 2 Review identified packets and begin filtering access to the infrastructure.

Once the packets filtered by the discovery ACL have been identified and reviewed, deploy an ACL with a permit any source to infrastructure addresses for the expected protocols.

As in Step 1, the **log** keyword can provide more information about the packets that match the `permit` entries. Using the **deny ip any** *your_ infrastructure_address_block* command at the end can help identify any unexpected packets destined to the infrastructure equipment. In case the ACL is deployed in transit networks, the last entry should be a **permit ip any any** statement to permit the flow of transit traffic. This ACL will provide basic protection and will allow network engineers to ensure that all required traffic is permitted.

Step 3 Restrict the range of source addresses.

Once you have a clear understanding of the protocols that must be permitted, further filtering can be performed to restrict the protocols to known or authorized source addresses. For example, for an ISP you can explicitly permit external BGP neighbors or specific GRE peer addresses.

This step narrows the risk of attack without breaking any services and allows you to apply granular control to sources that access your infrastructure equipment.

Step 4 (Optional): Limit the destination addresses within the iACL.

This final phase is meant to limit the range of destination addresses that will accept traffic for a given protocol. This helps restrict traffic more granularly.

Refer to Chapter 8, “Getting Started with Security Baseline,” for configuration examples.

Receive Access Control Lists

Receive Access Controls Lists (rACLs) is a feature designed to protect the Route Processor (RP) on high-end routers from unnecessary traffic that could potentially affect system performance. rACLs were originally introduced for the Cisco 12000 Series Routers, but are now available on other high-end routing platforms including the Cisco 7500 Series Routers and the Cisco 10000 Series Routers.

Simply put, a rACL is an access control list that controls the traffic sent by the various line cards to the RP on distributed architectures like the Cisco 1200 Series Routers. When configured, rACLs are first created on the RP, and then pushed to the line card CPUs. When packets enter the line cards, the packets are first sent to the line card CPU. Packets requiring processing by the RP are compared against the rACL before being sent to the RP. It should be note that rACLs apply to traffic destined to the RP only, and does not affect transit traffic.

rACLs are defined as standard or extended ACLs. They typically consist of permit statements allowing the protocols and sources that are expected by the RP, and may also include deny statements explicitly blocking unwanted traffic. Like other ACLs, rACLs have an implicit **deny ip any any** at the end.

rACL help mitigate attacks directed at the RP that are intended to overwhelm its capacity. RPs always have a limited capacity to process traffic delivered from the line cards. If a high volume of data requires punting traffic to the RP, this may overwhelm the RP, resulting in a denial of service (DoS) condition. Under such circumstances the CPU of the RP struggles to keep up with the packet examination and begins dropping packets, thereby flooding the input-hold and Selective Packet Discard (SPD) queues.

Under normal circumstances, most of the traffic handled by a router is in transit over the forwarding path. Only a small portion of the traffic needs to be sent to the RP over the receive path for further analysis. Examples of traffic that is directed to the router itself, and which is handled by the RP includes the following:

- Routing protocols
- Remote access protocols, such as SSH and telnet
- Network management traffic, such as SNMP
- Other protocols, such as ICMP, or IP options, might also require processing by the RP

Because rACLs allow only authorized traffic to be sent to the RP, they help mitigate attacks directed to the RP itself. It should be noted however, that rACLs are not effective mitigating attacks originated from trusted sources and based on trusted protocols (those permitted by the rACL entries).

By protecting the RP, rACLs help ensure router and network stability during attacks. For this reason, their deployment is recommended on all routers, but particularly on those facing the Internet or other external networks. As explained next in this document, Control Plane Policing (CoPP) extends the functionality of rACLs by incorporating rate limiting. In general, CoPP is preferred over rACLs, however rACLs are simpler to deploy and may be the best fit for users looking to avoid the more complex configuration of CoPP. rACLs can also be implemented as a first step to CoPP. The guidelines for migrating from rACL to CoPP are provided in *Infrastructure Protection on Cisco IOS Software-Based Platforms* at the following URL:

http://www.cisco.com/en/US/prod/collateral/iosswrel/ps8802/ps6970/ps1838/prod_white_paper0900aecd804ac831.pdf

As discussed previously in this document, having an adequate design of the address space facilitates deployment of security measures, in particular access control mechanisms like rACLs.

For more information about rACLs, refer to the following URL:

http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a00801a0a5e.shtml

rACL Recommended Deployment Methodology

Because rACLs filter traffic, before deploying rACLs, it is vital to gain an adequate understanding about the legitimate traffic destined to the RP. An rACL built without the proper understanding of the protocols and the devices involved might block critical traffic, potentially creating a denial of service (DoS) condition.

In some networks, determining the exact traffic profile needed to build the filters might be difficult. For this reason, this document recommends a conservative methodology for deploying rACLs using iterative ACL configurations to help identify and eventually filter traffic.

To deploy rACLs using this methodology, complete the following steps:

Step 1 Identify Protocols used in the network with a discovery ACL.

Start by deploying a discovery or classification rACL permitting all the commonly used protocols that access the RP. Appendix B, "Commonly Used Protocols in the Infrastructure," contains a list of commonly used protocols in the infrastructure.

This discovery rACL should have both source and destination addresses set to any. Logging can be used to develop a list of source addresses that match the protocol permit statements. In addition to the protocol permit statement, a permit any any log line at the end of the rACL can be used to identify other protocols that would be filtered by the rACL and might require access to the RP.

The objective is to determine the protocols and the specific network uses. Logging should be used for analysis to determine everything else that might be communicating with the router.

Note Although the log keyword provides valuable insight into the details of ACL hits, excessive hits to an ACL entry that uses the log keyword might result in an overwhelming number of log entries and possibly high router CPU utilization. Use the log keyword for short periods of time and only when needed to help classify traffic.

Step 2 Review identified packets and begin to filter access to the RP.

Once the packets filtered by the rACL in Step 1 have been identified and reviewed, implement an rACL with a permit any any statement for each of the expected protocols.

As in Step 1, the **log** keyword can provide more information about the packets that match the permit entries. Using deny any any log at the end can help identify any unexpected packet destined to the RP. This rACL will provide basic protection and allow network engineers to ensure that all required traffic is permitted.

The objective is to test the range of protocols that need to communicate with the router without having the explicit range of IP source and destination addresses.

Step 3 Restrict the range of source addresses.

Only allow addresses within your allocated address block as source addresses. For example, if you are using the 198.133.219.0/24 block for your network, then permit source addresses only from 198.133.219.0/24.

This step narrows the risk of attack without breaking any services. It also provides data points of devices and users from outside your address block that might be accessing your equipment. Traffic from all outside addresses will be dropped.

There may be some exceptions to the previous rule, for example an eBGP peer will require an exception because the permitted source addresses for the session will lie outside your address block.

This phase might be left on for a few days to collect data for the next phase of narrowing the rACL.

Step 4 Narrow the rACL permit statements to only allow known authorized source addresses.

Increasingly limit the source address to only permit sources that communicate with the RP.

Step 5 (Optional): Limit the destination addresses in the rACL.

This final phase is meant to limit the range of destination addresses that will accept traffic for a given protocol. This helps restrict traffic more granularly.

Refer to Appendix A, "Sample Configurations," for sample configurations.

.

Control Plane Policing (CoPP)

Control Plane Policing (CoPP) is a security infrastructure feature that protects the control plane of routers and switches by enforcing QoS policies that regulate the traffic processed by the main system CPU (route or switch processor). With CoPP, these QoS policies are configured to permit, block, or rate limit the packets handled by the main CPU. This helps protects the control plane of routers and switches from a range of attacks, including reconnaissance and direct DoS.

CoPP applies to packets handled by the main CPU, referred to as control plane traffic, and which typically include the following:

- Routing protocols
- Packets destined to the local IP address of the router
- Packets from network management protocols, such as SNMP
- Interactive access protocols, such as SSH and Telnet
- Other protocols, such as ICMP, or IP options, might also require handling by the switch CPU
  - Layer 2 packets such as BPDU, CDP, DOT1X, etc.

CoPP leverages the modular QoS command-line interface (MQC) for its policy configuration. MQC allows the separation of traffic into classes, and lets the user define and apply distinct QoS policies to each class. The QoS policies can be configured to permit all packets, drop all packets, or drop only those packets exceeding a specific rate limit.

CoPP is available on a wide range of Cisco platforms, which all deliver the same basic functionality. However, CoPP has been enhanced on some platforms to leverage the benefits of the particular hardware architectures. As a result, some platforms provide advanced forms of CoPP. Non-distributed platforms implement a centralized software-based CoPP model, while some distributed platforms provide enhanced versions of CoPP: distributed and hardware-based. In addition, as a result of the hardware differences, CoPP protocol support may vary depending on the platform. This document provides a generic description of CoPP. For detailed information on how CoPP is implemented on particular platforms, refer to the list of documents provided at Appendix C, "Related Documents."

Functionally, CoPP comes into play right after the switching or the routing decision, and before traffic is forwarded to the control plane. When CoPP is enabled, at a high level the sequence of events as follows:

Step 1 A packet enters the router/switch configured with CoPP on the ingress port.

Step 2 The port performs any applicable input port and QoS services.

Step 3 The packet gets forwarded to the router/switch processor.

Step 4 The router/switch processor makes a routing/switching decision, determining whether or not the packet is destined to the control plane.

Step 5 Packets destined for the control plane are processed by CoPP, and are dropped or delivered to the control plane according to each traffic class policy. Packets that have other destinations are forwarded normally.

Compared with rACLs, the rate-limiting capability of CoPP makes it more effective when dealing with DoS attacks against the control plane, in particular those based on authorized protocols and sources. rACLs only permit or deny traffic, and there are scenarios in which such a binary response is not sufficient. As an example, ICMP echo requests (pings) are commonly allowed for diagnostic purposes and should be permitted when used for their intended purpose. However, a large volume of ICMP echo-requests can overwhelm the RP and may be part of a DoS attack. CoPP can effectively handle this kind of situation by enforcing rate limiting policies per traffic class. For example, using MQC, you can define a traffic class to include ICMP echo requests and then drop packets exceeding the specified rate limit for this traffic class.

In general, CoPP is recommended on all routers and switches, but particularly on those facing the Internet or other external networks. Some users may still prefer to use rACL for simplicity, or as a first step toward implementing CoPP. For guidelines on how to migrate from rACL to CoPP, refer to *Infrastructure Protection on Cisco IOS Software-Based Platforms* at the URL below.

- For more information about CoPP, see the following:

Infrastructure Protection on Cisco IOS Software-Based Platforms

http://www.cisco.com/application/pdf/en/us/guest/products/ps1838/c1244/cdccont_0900aecd804ac831.pdf

- Infrastructure Protection on Cisco Catalyst 6500 and 4500 Series Switches

http://www.cisco.com/application/pdf/en/us/guest/netsol/ns171/c649/ccmigration_09186a0080825564.pdf

CoPP Traffic Classification

Because CoPP filters traffic, it is critical to gain an adequate level of understanding about the legitimate traffic destined to the RP or SP prior to deployment. CoPP policies built without proper understanding of the protocols, devices or required traffic rates involved may block critical traffic. This has the potential of creating a denial of service (DoS) condition. Determining the exact traffic profile needed to build the CoPP policies might be difficult in some networks. For this reason, this document describes a conservative methodology for deploying CoPP using iterative ACL configurations to help identify and to incrementally filter traffic.

Prior to developing an actual CoPP policy, required traffic must be identified and separated into different classes. Multiple classification schemes can be used, but one recommended methodology involves classifying traffic into distinct groups based on relative importance and traffic type. This section presents an example based on ten different classes, which provides great granularity and is suitable for real world environments. It is important to note, that, even though you can use this example as a reference, the actual number and type of classes needed for a given network may differ and should be selected based on local requirements, security policies, and a thorough analysis of baseline traffic.

This example defines the following nine traffic classes.

Border Gateway Protocol (BGP)

This class defines traffic that is crucial to maintaining neighbor relationships for BGP routing protocol, such as BGP keepalives and routing updates. Maintaining BGP routing protocol is crucial to maintaining connectivity within a network or to an ISP. Sites that are not running BGP would not use this class.

Interior Gateway Protocol (IGP)

This class defines traffic that is crucial to maintaining IGP routing protocols such as Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), and Routing Information Protocol (RIP). Maintaining IGP routing protocols is crucial to maintaining connectivity within a network.

Interactive Management

This class defines interactive traffic that is required for day-to-day network operations. This class would include light volume traffic used for remote network access and management. For example, telnet, Secure Shell (SSH), Network Time Protocol (NTP), Simple Network Management Protocol (SNMP) and Terminal Access Controller Access Control System (TACACS).

File Management

This class defines high volume traffic used for software image and configuration maintenance. This class would include traffic generated for remote file transfer. For example, Trivial File Transfer Protocol (TFTP), and File Transfer Protocol (FTP).

Reporting

This class defines traffic used for generating network performance statistics for reporting. This class would include traffic required for using Service Assurance Agent (SAA) to generate ICMP with different DSCP settings in order to report on response times within different QoS data classes.

Monitoring

This class defines traffic used for monitoring a router. This kind of traffic should be permitted but should never be allowed to pose a risk to the router. With CoPP, this traffic can be permitted but limited to a low rate. Examples would include packets generated by ICMP echo requests (`ping`) and the `traceroute` command.

Critical Applications

This class defines application traffic that is crucial to a specific network. The protocols that might be included in this class include generic routing encapsulation (GRE), Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), Session Initiation Protocol (SIP), Data Link Switching (DLSw), Dynamic Host Configuration Protocol (DHCP), IPSec, and Multicast traffic.

Layer 2 Protocols

This class defines traffic used for Address Resolution Protocol (ARP). Excessive ARP packets can potentially monopolize RP resources, starving other important processes. CoPP can be used to rate limit ARP packets to prevent this. Currently, ARP is the only Layer 2 protocol that can be specifically classified using the `match` command.

Undesirable

This explicitly identifies unwanted or malicious traffic that should be dropped and denied access to the RP. For example, this class could contain packets from a well-known worm. This class is particularly useful when specific traffic destined to the router should always be denied rather than be placed into a default category. Explicitly denying traffic allows you to collect rough statistics on this traffic using `show` commands and thereby offers some insight into the rate of denied traffic.

Default

This class defines all remaining traffic destined to the RP that does not match any other class. MQC provides the Default class so you can specify how to treat traffic that is not explicitly associated with any other user-defined classes. It is desirable to give such traffic access to the RP but at a highly reduced rate.

With a default classification in place, statistics can be monitored to determine the rate of otherwise unidentified traffic destined to the control plane. Once this traffic is identified, further analysis can be performed to classify it. If needed, the other CoPP policy entries can be updated to account for this traffic.

CoPP Recommended Deployment Methodology

To implement the conservative methodology recommended for deploying CoPP, complete the following steps:

Step 1 Determine the classification scheme for your network.

Identify the known protocols that access the RP and divide them into categories using the most useful criteria for your specific network. For example, the ten categories given in the example in this section (BGP, IGP, Interactive Management, File Management, Reporting, Critical Applications, Layer 2 Protocols, Undesirable, and Default) use a combination of relative importance and traffic type. Select a scheme suited to your specific network, which may require a larger or smaller number of classes.

Step 2 Configure classification ACLs.

Develop an ACL for each class identified in Step 1, including one for the Default class.

Configure each ACL to permit all known protocols in its class that require access to the RP. At this point, each ACL entry should have both source and destination addresses set to **any**. In addition, the ACL for the Default class should be configured with a single entry: **permit ip any any**. This will match traffic not explicitly permitted by entries in the other ACLs.

Once the ACLs have been configured, create a **class-map** for each class defined in Step 1, including one for the Default class. Then assign each ACL to its corresponding **class-map**.

Note In this step you should create a separate **class-map** for the default class, rather than using the **class-default** available on some platforms. Creating a separate **class-map**, and assigning a **permit ip any any** ACL, will allow you to identify traffic not yet classified as part of another class.

Each class map should then be associated with a **policy-map** that permits all traffic, regardless of classification. The policy for each class should be set as **conform-action transmit exceed-action transmit**.

Step 3 Review the identified traffic and adjust the classification.

Ideally, the classification performed in Step 1 identified all required traffic destined to the router. However, realistically, not all required traffic will be identified prior to deployment and the **permit ip any any** entry in the class Default ACL will log a number of packet matches. Some form of analysis will be required to determine the exact nature of the unclassified packets.

Use the **show access-lists command** to see the entries in the ACLs that are in use, and to identify any additional traffic sent to the RP. To analyze the unclassified traffic you can use one of the following techniques:

- General ACL classification as described in Characterizing and Tracing Packet Floods Using Cisco Routers, available at the following URL:
- http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080149ad6.shtml
- Packet analyzers
- rACLs

Once traffic has been identified, adjust the class configuration accordingly. Remove the ACL entries for those protocols that are not used. Add a **permit any any** entry for each protocol just identified.

Step 4 Restrict a macro range of source addresses.

Refine the classification ACLs, by only allowing the full range of the allocated address block to be permitted as the source address. For example, if the network has been allocated 172.26.0.0/16, then permit source addresses from 172.26.0.0/16 where applicable.

This step provides data points for devices or users from outside the assigned address block that might be accessing the equipment. For example, an external BGP (eBGP) peer will require an additional ACL entry because the permitted source addresses for the session will lay outside the assigned address block. This phase might be left on for a few days to collect data for the next phase of narrowing the ACL entries.

Step 5 Narrow the ACL permit statements to authorized source addresses.

Increasingly limit the source address in the classification ACLs to only permit sources that communicate with the RP. For instance, only known network management stations should be permitted to access the SNMP ports on a router.

Step 6 Refine CoPP policies by implementing rate limiting.

Use the **show policy-map control-plane** command to collect data about the actual policies in place. Analyze the packet count and rate information and develop a rate limiting policy accordingly.

At this point, you may decide to remove the **class-map** and ACL used for the classification of default traffic. If so, you should also replace the previously defined policy for the Default class by the **class-default** policy.

Refer to Appendix A, “Sample Configurations,” for sample configurations.

Control Plane Protection (CPP)

Control Plane Protection (CPP) is a security feature that extends the policing functionality provided by the software-based Control Plane Policing (CoPP) feature. The CoPP feature controls the rate in which control plane traffic is sent to he Route Processor in Cisco IOS software-based devices. Control Plane Protection extends this policing functionality by dividing the Control Plane into three control plane sub-interfaces and allowing the enforcement of separate rate-limiting policies. In addition, CPP incorporates port-filtering and queue-thresholding. Port-filtering is a mechanism for the early dropping of packets that are directed to closed or non-listened IOS TCP/UDP ports. Queue-thresholding is a mechanism that limits the number of packets per protocol hold in the control-plane input queue, preventing the input queue from being overwhelmed by any single protocol traffic.

CPP is a feature that extends the policing functionality of the software-based CoPP by providing an additional layer of protection to the control plane. With CPP, the first layer of protection is provided by CoPP at an aggregate level by controlling all packets destined to the control plane. Once traffic is processed by CoPP is then handled to CPP, the second layer of protection, and which divides the traffic into three categories. Each category is processed by a control plane sub-interface with independent rate-limiting policies. This dual layer of protection provides a control hierarchy that allows for finer policy definition and enforcement.

Note Control Plane Protection is only available on platforms that support software-based CoPP. This feature is currently not available on platforms with hardware-based or distributed CoPP.

The three control plane sub-interfaces implemented by Control Plane Protection are:

- **Control-plane host subinterface—**This interface handles all control-plane IP packets that are destined to any of the IP addresses configured on the router interfaces. Examples of traffic falling in this category include tunnel termination traffic, management traffic or routing protocols such as SSH, SNMP, BGP, OSPF, and EIGRP. All host traffic terminates on and is processed by the router.
- **Control-plane transit subinterface—**This subinterface receives all IP packets that are software switched by the route processor. This means packets that are not directly destined to the router itself but rather traffic traversing through the router and that require process switching.
- **Control-plane CEF-exception subinterface—**This control-plane subinterface receives all IP packets that are either redirected as a result of a configured input feature in the CEF packet forwarding path for process switching or directly enqueued in the control plane input queue by the interface driver (i.e. ARP, L2 Keepalives and all non-IP host traffic).

In addition, CPP enhances the protection of the control-plane host subinterface by implementing Port-filtering and Queue-thresholding. Port-filtering is a feature that can only be applied to the control-plane host subinterface and that automatically drops packets directed toward closed or non-listened UDP/TCP ports on the router. Queue-thresholding is another feature that can only be applied to the control-plane host subinterface and that limits the number of unprocessed packets per protocol, preventing the input queue from being overwhelmed by any single protocol traffic.

At a very high level the sequence of events with Control Plane Protection is as follows:

Step 1 A packet enters the router configured with CoPP on an ingress interface.

Step 2 The interface performs the basic input port and QoS services.

Step 3 The packet gets forwarded to the router processor.

Step 4 The router processor makes a routing decision, determining whether or not the packet is destined to the control plane.

Step 5 Packets destined for the control plane are processed by Aggregate CoPP, and are dropped or forward to the Control Plane Path according to the polices for each traffic class. Packets that have other destinations are forwarded normally.

Step 6 Packets sent to the Control Plane Path are intercepted by the Control Plane Protection traffic classifier, which classifies the packets into the corresponding control-plane subinterfaces.

Step 7 Packets received by each control-plane subinterface are dropped or forward to the Control Plane global input queue according to the configured policies.

Step 8 In addition, packets sent to the control-plane host subinterface can be dropped or forwarded according to the Port-filter and Queue-thresholding policies before they are sent to the global input queue.

Similar to CoPP, CPP helps protect the RP of Cisco IOS software-based routers by filtering unwanted traffic and by rate-limiting the traffic expected by the control plane. This shields the control plane from traffic that might be part of DoS or other attacks, helping maintain network stability even during attack conditions.

CPP ability to divide the control plane traffic and rate-limit each traffic type individually, gives you greater traffic control for attack mitigation. Port-filtering and Queue-thresholding also provide for a more advanced attack protection. On one hand, Port-filtering shields the RP from packets directed to closed or non-listened TCP/UDP ports, mitigating attacks attempting to spoof legitimate traffic permitted by CoPP. On the other hand, Queue-thresholding limits protocol queue usage mitigating attacks designed to overwhelm the input queue with the flooding of a single protocol.

CPP is recommended on all software-based IOS platforms, where hardware-based CoPP is not available. CPP is particularly useful on routers facing the Internet or other external networks.

Control Plane Protection Recommended Deployment Methodology

By protecting the RP, CPP helps ensure router and ultimately network stability during an attack. For this reason, CPP should be deployed on all software-based routers as a key protection mechanism.

Because CPP filters traffic, it is critical to gain an adequate level of understanding about the legitimate traffic destined to the RP prior to deployment. CPP policies built without proper understanding of the protocols, devices or required traffic rates involved may block critical traffic. This has the potential of creating a denial of service (DoS) condition. Determining the exact traffic profile needed to build the CPP policies might be difficult in some networks. For this reason, this document describes a conservative methodology for deploying Control Plane Protection using iterative ACL configurations to help identify and to incrementally filter traffic.

Defining CPP policies requires the use of the MQC CLI following the same procedures explained for Control Plane Policing. For examples illustrating the use of MQC CLI to configure traffic classes and policies, please refer to the CoPP section of this document.

For configuration examples, refer to Appendix A, “Sample Configurations.”

Port Security

An attacker can mount a DoS attack against infrastructure devices by using MAC flooding to cause MAC address table exhaustion, as well as other Layer 2 Content Addressable Memory (CAM) overflow attacks. This type of attack can be addressed with a Cisco feature called Port Security. Port Security helps mitigate MAC flooding and other Layer 2 CAM overflow attacks by restricting the MAC addresses that are allowed to send traffic on a particular port. Once Port Security is enabled on a port, only packets with a permitted source MAC address are allowed to pass through the port. A permitted MAC address is referred to as a secure MAC address.

Port Security builds a list of secure MAC addresses in one of two ways, configurable on a per-interface basis:

- Dynamic learning of MAC addresses
  - Defines a maximum number of MAC addresses that will be learnt and permitted on a port.
  - Useful for dynamic environments, such as at the access edge.
- Static configuration of MAC addresses
  - Defines the static MAC addresses permitted on a port.
  - Useful for static environments, such as a server farm, a lobby, or a Demilitarized Network (DMZ).

It is possible to combine these two options on a single interface by defining a maximum number of MAC addresses to be permitted, along with some static MAC addresses. In this scenario, the static MAC addresses will count towards the maximum number of MAC addresses permitted. This may be used, for instance, to ensure that only a single, statically defined host is permitted to communicate on a specific port, such as in a lobby.

A security violation occurs when either:

- The maximum number of secure MAC addresses is reached on a secure port and the source MAC address of the ingress traffic is different from any of the identified secure MAC addresses.
- Traffic with a secure MAC address that is configured or learned on one secure port appears in another secure port in the same VLAN.

In Cisco IOS, the action to be taken upon a security policy violation is configurable based on the three options shown in Table 4-3.

Table 4-3 Actions for Security Policy Violations

Port Security Violation Mode	Action Upon Security Violation	Notification Events	Port Re-activation Following Security Violation
Protect	Packets with an unknown source MAC address are dropped	None	Dynamic MAC address learning resumes upon the number of learnt MAC addresses dropping below the configured maximum number.
Restrict[1]	Packets with an unknown source MAC address are dropped	Syslog message generated Security violation counter increments SNMP trap generated (if enabled)	Dynamic MAC address learning resumes upon the number of learnt MAC addresses dropping below the configured maximum number.
Shutdown (default)[2]	Interface is error-disabled and the port disabled	Syslog message generated Security violation counter increments SNMP trap generated (if enabled)	The port can only be re-activated through manual intervention[3].

[1] Port Security restrict mode may have a significant impact on device CPU. Note that the Port Security restrict violation mode will impact the CPU when an attack is in progress. It is thus recommended that the performance impact of this feature and its possible implications are carefully tested and considered prior to deployment

[2] Port Security shutdown mode is the default security violation action.

[3] With Port Security shutdown mode, upon occurrence of a security violation, a port can only be re-activated by either entering the global configuration **errdisable recovery cause psecure-violation** command or by shutting down and re-enabling the port with **shutdown** and **no shutdown** commands.

Typical deployment scenarios consist of:

- A dynamic environment, such as an access edge, where a port may have Port Security enabled with the maximum number of MAC addresses set to one, enabling only one MAC address to be dynamically learnt at any one time, and a protect response action.
- A static, controlled environment, such as a server farm or a lobby, where a port may have Port Security enabled with the server or lobby client MAC address statically defined and the more severe response action of shutdown.

Note The static configuration and administration of large numbers of MAC address can present an operational challenge that should be balanced against the security risks.

A VoIP deployment, where a port may have Port Security enabled with the maximum number of MAC addresses defined as two, since two MAC addresses are required per port, one for the workstation and one for the phone. In addition, it is generally recommended that the security violation action be set to restrict so that the port is not entirely taken down when a violation occurs. However, care should be taken due to the possible CPU impact of restrict mode.

Port Security is supported on trunk ports but requires some specific configuration rules to be followed. Trunk Port Security allows the configuration of Port Security related parameters on a per VLAN-port basis, applying policy according to a specific VLAN on a specific port.

Port Security Configuration

In Cisco IOS, Port Security can be enabled on an interface using the command `switchport port-security`. The example below shows dynamic Port Security, restricted to two MAC addresses, being applied to an interface, with a security violation mode of restrict, such as may be deployed on a VoIP-enabled port.

```
Router(config)# interface gigabitethernet0/1
Router(config-if)# switchport port-security maximum 2
Router(config-if)# switchport port-security violation restrict
Router(config-if)# switchport port-security
```

The example below illustrates how a port can be restricted for use by only one specific host, with the defined MAC address, such as may be employed in a lobby environment.

```
Router(config)# interface gigabitethernet0/2
Router(config-if)# switchport port-security maximum 1
Router(config-if)# switchport port-security mac-address 1000.2000.3000
Router(config-if)# switchport port-security violation restrict
Router(config-if)# switchport port-security
```

An additional configuration option called ‘sticky learning’ is also available. Sticky Port Security retains learnt MAC addresses across reboots, though it is not available on all switches. When sticky learning is enabled, the interface adds all MAC addresses that are dynamically learned to the running configuration and converts these addresses to sticky secure MAC addresses.

In Cisco IOS, sticky Port Security can be enabled on an interface using the command **`switchport port-security mac-address sticky`**.

```
Router(config)# interface gigabitethernet0/1
Router(config-if)# switchport port-security mac-address sticky
```

Port Security aging is also configurable for both static and dynamic addresses, allowing the aging timers and aging types to be defined. The timer is defined in minutes and can be configured as an absolute or as an inactivity timeout.

In Cisco IOS, Port Security aging can be enabled on an interface using the command `switchport port-security aging`. The example below shows an inactivity aging time of two minutes being applied to an interface.

```
Router(config)# interface gigabitethernet0/1
Router(config-if)# switchport port-security aging time 2
Router(config-if)# switchport port-security aging type inactivity
```

Port Security Logging

The SNMP logging of Port Security policy violations can be enabled using the following command:

```
snmp-server enable traps port-security
```

SNMP trap rate-limiting can also be enabled to reduce the load on a device during an attack using the following command:

```
snmp-server enable traps port-security trap-rate <max number of traps per second>
```

Note An SNMP trap will only be sent if a security policy violation mode of restrict or shutdown is enabled on an interface.

For more information on the **switchport port-security** command on the Catalyst 3560, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_37_se/command/reference/cli3.html#wp1948361

For more information on how to configure Port Security on a Catalyst 6500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SXF/native/configuration/guide/port_sec.html

For more information on how to configure Port Security on Catalyst 4500 running Cisco IOS, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/31sg/configuration/guide/port_sec.html

Redundancy

Networks are built out of numerous hardware and software components that may fail or that may be subject to attacks. Implementing redundant designs helps eliminate single points of failure, improving the availability of the network and making it more resistant to attacks. There are different ways one can implement redundancy, from deploying simple backup interfaces up to building complete redundant topologies. Certainly, making every single component redundant is costly; therefore design redundancy where most needed and according to the unique requirements of your network.

Cisco products offer a wide range of options for redundancy:

- Backup interfaces
- Element redundancy
- Standby devices
- Topological Redundancy

Backup Interfaces

Cisco routers and other Cisco products allow the configuration of backup interfaces. A backup interface is an interface that remains in standby mode until the primary interface fails or goes down. The backup interface can be a physical interface such as a Basic Rate Interface (BRI), or an assigned backup dialer interface to be used in a dialer pool. When in standby mode, the backup interface remains shutdown and any routes associated with it do not appear in the routing table. The backup interface is brought up when the router detects that the primary interface goes down. Backup interfaces are implemented in pairs, and are commonly used to back up ISDN BRI connections, async lines and leased lines.

One benefit of backup interfaces is that they are independent of routing protocols; hence their operation is not conditioned by routing protocol convergence, route stability and so on. However, depending on the encapsulation used, the router may not detect when an interface goes down and in consequence the backup interface may not be brought up. For example, with a Frame Relay connection, the line protocol may not go down when a particular PVC/DLCI goes down. Since the router cannot detect the failure, the backup link may not be activated.

In Cisco IOS, backup interfaces are configured with the **backup interface** command. The following example sets serial 1 as the backup line to serial 0:

```
interface serial 0
 backup interface serial 1
```

For more information on the **backup interface** command, refer to the following URL:

http://www.cisco.com/en/US/products/ps6350/products_command_reference_chapter09186a008044377b.html#wp1078001

In addition to backup interfaces, Cisco IOS offers other features useful for implementing link redundancy. Reliable Static Routing Backup Using Object Tracking is one feature that allows to reliably backup PPPoE and IPSec tunnels. This feature relies on ICMP pings to monitor the tunnel, and when the primary gateway becomes unreachable thorough the primary channel, a DDR connection is initiated from an alternative port. In this way, Reliable Static Routing Backup Using Object Tracking ensures reliable backup in the case of several catastrophic events, such as Internet circuit failure or peer device failure. In addition, this feature is compatible with both preconfigured static routes and Dynamic Host Configuration Protocol (DHCP) configurations.

For more information on *Backup Interface for Reliable Static Routing Backup Using Object Tracking*, refer to the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps5413/products_feature_guide09186a00801d862d.html

Element Redundancy

Some modular platforms allow the configuration of redundant Route Processors and other critical components, helping maintain the overall system and network availability. Cisco IOS routers provide the following element redundancy features:

- **High System Availability (Cisco 7500)—**HSA allows you to install two Route Processors in a single router to improve system availability. This feature is available only on Cisco 7500 series routers. Supporting two RPs in a router provides the most basic level of increased system availability through a "cold restart" feature. A cold restart means that when one RP fails, the other RP reboots the router. Thus, the router is never in a failed state for very long, thereby increasing system availability.
- **High Availability NPE Redundancy (Cisco 7300)—**Route processing redundancy is a feature currently available on the Cisco 7304 router. With two NPE-G100s installed in a router, the feature provides the most basic level of increased system availability through a "partial bootup" feature on the standby NPE-G100. A "partial bootup" means that when the active NPE-G100 fails or a fatal error is detected on the NPE-G100, the standby NPE-G100 will complete booting and take control

of the line cards. This minimizes the time that the router is in a failed state, thereby increasing system availability. Switchover takes approximately one minute. Configuration syncing of startup configuration (only) and ROMmon environmental variables are supported.

- **Route Processor Redundancy (RPR)—** RPR is an alternative mode to HSA and allows Cisco IOS software to be booted on the standby processor prior to switchover (a "cold boot"). In RPR, the standby RP loads a Cisco IOS image at boot time and initializes itself in standby mode; however, although the startup configuration is synchronized to the standby RP, system changes are not. In the event of a fatal error on the active RP, the system switches to the standby processor, which reinitializes itself as the active processor, reads and parses the startup configuration, reloads all of the line cards, and restarts the system.
- **Route Processor Redundancy Plus—** In RPR+ mode, the standby RP is fully initialized. For RPR+ both the active RP and the standby RP must be running the same software image. The active RP dynamically synchronizes startup and the running configuration changes to the standby RP, meaning that the standby RP need not be reloaded and reinitialized (a "hot boot"). Additionally, on the Cisco 10000 and 12000 series Internet routers, the line cards are not reset in RPR+ mode. This functionality provides a much faster switchover between the processors. Information synchronized to the standby RP includes running configuration information, startup information (Cisco 7304, Cisco 7500, Cisco 10000, and Cisco 12000 series devices), and changes to the chassis state such as online insertion and removal (OIR) of hardware. Line card, protocol, and application state information is not synchronized to the standby RP.
- **Stateful Switchover (SSO)—**SSO mode provides all the functionality of RPR+ in that Cisco IOS software is fully initialized on the standby RP. In addition, SSO supports synchronization of line card, protocol, and application state information between RPs for supported features and protocols (a "hot standby"). During switchover, system control and routing protocol execution are transferred from the active to the standby RP. Switchover may be due to a manual operation (CLI-invoked) or to a software- or hardware-initiated operation (hardware or software fault induced).

For more information on how to configure High System Availability on your Cisco 7500 series routers, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_0/configfun/configuration/guide/fc_hsa.html

For more information on how to configure High Availability NPE Redundancy on your Cisco 7300 series routers, refer to the following URL:

http://www.cisco.com/en/US/docs/routers/7300/install_and_upgrade/7304/7304_fru/7304_npe_icg/O3613h.html

For more information on how to configure Route Processor Redundancy on your Cisco 7500 series routers, refer to the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1612/products_white_paper09186a00800809c8.shtml

For more information on how to configure Route Processor Redundancy Plus on your Cisco routers, refer to the following URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a008045526b.html

For more information on how to configure Stateful Switchover on your Cisco routers, refer to the following URL:

http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_white_paper09186a00801ce6f9.shtml

Standby Devices

Cisco products offer a range of failover mechanisms and redundancy protocols that allow you to deploy redundant devices, increasing system and network availability. In a typical scenario, devices are deployed in pairs, but may also be set up in groups.

Failover mechanisms can be Active/Standby or Active/Active, and Stateless or Stateful:

- **Active/Standby Failover—**In this configuration two pairs of devices are deployed. One of the devices is active, and as such it handles all the network traffic. The other device remains idle in standby mode. The standby device does not process network traffic until a failure occurs on the active device. When a failover occurs, the standby device becomes the active unit. It assumes the IP and MAC addresses of the previously active unit. Because the other devices on the network do not see any changes in the IP or MAC addresses, ARP entries do not change or time out anywhere on the network.
- **Active/Active Failover—**In an Active/Active failover configuration, both devices are active and handle network traffic. Active/Active allows load balancing traffic. This type of failover typically requires the separation of traffic in different contexts, where each unit is configured as primary for some contexts and standby for others. In some platforms, Active/Active Failover can be implemented by the means of packet or session load balancing.
- **Stateless Failover—** In this mode of failover, all active connections are dropped when a failover occurs. Clients need to reestablish connections when the new active unit takes over.
- **Stateful Failover:—**In this mode, the active unit in the failover pair continually passes per-connection state information to the standby unit. After a failover occurs, the same connection information is available at the new active unit. Supported end-user applications are not required to reconnect to keep the same communication session.

For more information on how to configure Firewall Stateful Failover on your Cisco routers, refer to the following URL:

http://www.cisco.com/en/US/products/ps6441/products_feature_guide09186a00806106ea.html

For more information on how to configure IPSec Stateful Failover on your Cisco routers, refer to the following URL:

http://www.cisco.com/en/US/products/ps6550/products_white_paper09186a0080116d4c.shtml

For more information on how to configure Failover on your ASA appliances, refer to the following URL:

http://www.cisco.com/en/US/docs/security/asa/asa72/configuration/guide/failover.html

For more information on how to configure Failover on your FWSM, refer to the following URL:

http://www.cisco.com/en/US/docs/security/fwsm/fwsm23/configuration/guide/failover.html

Cisco routers also support First Hop Redundancy Protocols such as HSRP, VRRP and GLBP. These protocols are designed to allow for transparent fail-over at the first-hop IP router. With these protocols, two or more routers are set up together in a group, sharing a single IP address, the virtual IP address. The virtual IP address is configured in each end user's workstation as a default gateway address and is cached in the host's ARP cache. One router in the group is elected as the active router, and it is responsible for handling all traffic sent to the virtual IP address. In the event the active router fails, one of the standby routers in the group takes over.

First Hop Redundancy Protocols:

- **Hot Standby Router Protocol (HSRP)—**HSRP provides high network availability by providing first-hop routing redundancy for IP hosts on Ethernet, Fiber Distributed Data Interface (FDDI), Bridge-Group Virtual Interface (BVI), LAN Emulation (LANE), or Token Ring networks configured with a default gateway IP address. HSRP is used in a group of routers for selecting an active router and a standby router. In a group of router interfaces, the active router is the router of choice for routing packets; the standby router is the router that takes over when the active router fails or when preset conditions are met. When HSRP is configured on a network segment, it provides a virtual MAC address and an IP address that is shared among a group of routers running HSRP. The address of this HSRP group is referred to as the virtual IP address. One of these devices is selected by the protocol to be the active router. The active router receives and routes packets destined for the MAC address of the group. For n routers running HSRP, n + 1 IP and MAC addresses are assigned. HSRP detects when the designated active router fails, at which point a selected standby router assumes control of the MAC and IP addresses of the Hot Standby group. A new standby router is also selected at that time. HSRP uses a priority mechanism to determine which HSRP configured router is to be the default active router. Devices that are running HSRP send and receive multicast User Datagram Protocol (UDP)-based hello messages to detect router failure and to designate active and standby routers. When the active router fails to send a hello message within a configurable period of time, the standby router with the highest priority becomes the active router. The transition of packet forwarding functions between routers is completely transparent to all hosts on the network. In addition, HSRP supports MD5 algorithm authentication to protect against spoofing.
- **Virtual Router Redundancy Protocol (VRRP)—**Election protocol that dynamically assigns responsibility for one or more virtual routers to the VRRP routers on a LAN, allowing several routers on a multiaccess link to utilize the same virtual IP address. A VRRP router is configured to run the VRRP protocol in conjunction with one or more other routers attached to a LAN. In a VRRP configuration, one router is elected as the virtual router master, with the other routers acting as backups in case the virtual router master fails. The LAN clients can then be configured with the virtual router as their default gateway. The virtual router, representing a group of routers, is also known as a VRRP group. VRRP is supported on Ethernet, Fast Ethernet, BVI, and Gigabit Ethernet interfaces, and on MPLS VPNs and VLANs. You can configure VRRP in such a way that traffic to and from LAN clients can be shared by multiple routers, thereby sharing the traffic load more equitably among available routers. In addition, VRRP supports MD5 algorithm authentication to protect against spoofing.
- **Gateway Load Balancing Protocol (GLBP)—**GLBP provides automatic router backup for IP hosts configured with a single default gateway on an IEEE 802.3 LAN. Multiple first hop routers on the LAN combine to offer a single virtual first hop IP router while sharing the IP packet forwarding load. Other routers on the LAN may act as redundant GLBP routers that will become active if any of the existing forwarding routers fail. GLBP performs a similar function for the user as HSRP and VRRP. HSRP and VRRP allow multiple routers to participate in a virtual router group configured with a virtual IP address. HSRP and VRRP also allowed for load balancing by configuring multiple virtual router groups and by configuring different default gateways on the LAN clients. Unlike HSRP and VRRP, GLPB allows load balancing using a single virtual IP address, without having to configure different default gateways. To that end, the single virtual IP address is associated with multiple virtual MAC addresses. The forwarding load is shared among all routers in a GLBP group rather than being handled by a single router while the other routers stand idle. Each host is configured with the same virtual IP address, and all routers in the virtual router group participate in forwarding packets. GLBP members communicate between each other through hello messages sent every 3 seconds to the multicast address 224.0.0.102, User Datagram Protocol (UDP) port 3222 (source and destination). In addition, GLBP supports MD5 algorithm authentication to protect against spoofing.

For more information on using HSRP for Fault-Tolerant IP Routing, refer to the following URL:

http://www.cisco.com/en/US/tech/tk1330/technologies_design_guide_chapter09186a008066670b.html

For more information on using VRRP for Fault-Tolerant IP Routing, refer to the following URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a008042fbd9.html

For more information on using GLBP for Fault-Tolerant IP Routing, refer to the following URL:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a008042fb97.html

Topological Redundancy

Building networks with redundant links and devices help increase the overall availability of the network, making it more resistant to attacks. Topological redundancy can be implemented at both the network as well as the data link level. In both cases, the redundancy strategy depends on the dynamic capabilities of the network to recover from failure. At the network level this is done by using dynamic routing protocols like EIRP and OSPF, while at the data link level this is achieved by using the spanning tree protocol.

When implementing a redundant topology, it is critical to ensure that the resulting topology is consistent with the security policies and control mechanisms in place. In particular, no redundant path must bypass any of the controls in place.

The following documents provide guidance in the design of redundant topologies:

High Availability Campus Network Design—Routed Access Layer using EIGRP or OSPF

http://www.cisco.com/application/pdf/en/us/guest/netsol/ns432/c649/ccmigration_09186a0080811468.pdf

Data Center High Availability Clusters Design Guide

http://www.cisco.com/application/pdf/en/us/guest/netsol/ns500/c649/ccmigration_09186a00807c4528.pdf

Device Resiliency and Survivability Summary

Table 4-4 summarizes the best practices explained in this document and lists the threat they help mitigate.

Table 4-4 Device Resiliency and Survivability Summary

Security Feature/Technique	Attacks Mitigated	Best Used At
Disabling Unnecessary Services	Unauthorized access, reconnaissance, DoS based on unauthorized protocols, amplification attacks, access control bypass	All routers and switches
iACL	Unauthorized access, reconnaissance, DoS based on unauthorized protocols	Internet and other network edges, administrative edges
rACL	Unauthorized access, reconnaissance, DoS based on unauthorized protocols	All routers, primarily at Internet edge. rACL simplicity is prefer over CoPP.
CoPP	Unauthorized access, reconnaissance, DoS based on unauthorized protocols, DoS based on authorized protocols, Distributed DoS	All devices that provide hardware-based CoPP
Control Plane Protection	Unauthorized access, reconnaissance, DoS based on unauthorized protocols, DoS based on authorized protocols, Distributed DoS	Software only IOS devices
Redundancy	DoS, Distributed DoS	Critical routers and switches

CHAPTER 5

Network Telemetry

In order to operate and ensure availability of a network, it is critical to have visibility and awareness into what is occurring on the network at any one time. Network telemetry offers extensive and useful detection capabilities which can be coupled with dedicated analysis systems to collect, trend and correlate observed activity.

Baseline network telemetry is both inexpensive and relatively simple to implement. This section highlights the baseline forms of telemetry recommended for network infrastructure devices, including:

- Time Synchronization
- Local Device Traffic Statistics
- System Status Information
- CDP Best Common Practices
- Syslog
- SNMP
- ACL Logging
- Accounting
- Archive Configuration Change Logger
- Packet Capture

More information on network telemetry and the critical role it plays in security can be found in the whitepaper *How to Build a Cisco Security Operations Center.* This paper provides an overview of the principles behind security operations, along with guidance on how to build a security operations center. The whitepaper is available at the following URL:

http://www.cisco.com/en/US/solutions/collateral/ns341/ns524/ns546/ns310/net_implementation_white_paper0900aecd80598c16.html

CSF Methodology Assessment

The results of applying the CSF methodology for baseline network telemetry are presented in the table below and highlight the technologies and features identified and integrated in Network Security Baseline.

Visibility and Awareness

Table 5-1 CSF Methodology Assessment—Visibility and Awareness

Identify	Monitor	Correlate
• CDP • SNMP • Syslog	• NTP • Local device statistics • System Status Information – Memory/CPU/processes – CPU and memory threshold notification • CDP Best Common Practices • Logging – Syslog – SNMP – Accounting – Configuration change notification and logging • Packet capture – SPAN/RSPAN – Copy/capture VACLs	

Control and Containment

No control and containment features for CSF methodology.

Time Synchronization

When implementing network telemetry, it is important that dates and times are both accurate and synchronized across all network infrastructure devices. Without time synchronization, it is very difficult to correlate different sources of telemetry.

Enabling Network Time Protocol (NTP) is the most common method of time synchronization.

General best common practices for NTP include:

- A common, single time zone is recommended across an entire network infrastructure in order to enable the consistency & synchronization of time across all network devices.
- The time source should be from an authenticated, limited set of authorized NTP servers.

Detailed information on NTP and NTP deployment architectures is available in the *Network Time Protocol: Best Practices White Paper* at the following URL:

http://www.cisco.com/warp/public/126/ntpm.pdf

Timestamps and NTP Configuration

In Cisco IOS, the steps to enable timestamps and NTP include:

Step 1 Enable timestamp information for debug messages.

Step 2 Enable timestamp information for log messages.

Step 3 Define the network-wide time zone.

Step 4 Enable summertime adjustments.

Step 5 Restrict which devices can communicate with this device as an NTP server.

Step 6 Restrict which devices can communicate with this device as an NTP peer.

Step 7 Define the source IP address to be used for NTP packets.

Step 8 Enable NTP authentication.

Step 9 Define the NTP servers.

Step 10 Define the NTP peers.

Step 11 Enable NTP to update the device hardware clock

The Cisco IOS commands to achieve the above steps are provided below.

Timestamp information for debug messages can be enabled with the following global configuration command:

```
service timestamps debug datetime localtime show-timezone msec
```

Timestamp information for log messages can be enabled with the following global configuration command:

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

The network-wide time zone, shown in this example as PST, can be enabled with the following global configuration command:

```
clock timezone EST -5
```

Summertime adjustments, shown this example for PDT, can be enabled with the following global configuration command:

```
clock summer-time EDT recurring
```

A list of allowed NTP servers and peers can be enforced with an ACL:

```
access-list 10 remark ACL for NTP Servers and peers
access-list 10 permit <NTPserver1>
access-list 10 permit <NTPserver2>
access-list 10 permit <NTPpeer1>
access-list 10 deny any log
!
ntp access-group peer 10
```

NTP clients can be restricted with an ACL:

```
access-list 15 remark ACL for NTP clients
access-list 15 permit <Client1>
access-list 15 permit <Client2>
access-list 15 deny any log
```

```
!
ntp access-group serve-only 15
```

The source IP address to be used for NTP packets can be defined with the following global configuration command:

```
ntp source <Loopback or OOB interface>
```

The first step in enabling NTP authentication is to define an MD5 key to be used for NTP transactions:

```
ntp authentication-key <key#> md5 <strong8charkey>
```

The keys to be accepted for NTP authentication are subsequently defined with the following command:

```
ntp trusted-key <key#>
```

NTP authentication is enforced with the following global configuration command:

```
ntp authenticate
```

NTP is used to update the device hardware clock with the following global configuration command:

```
ntp update-calendar
```

The NTP servers are defined with the following global configuration command:

```
ntp server <NTPserver1>
ntp server <NTPserver2>
```

Any NTP peers are defined with the following global configuration command:

```
ntp peer <NTPpeer1>
ntp peer <NTPpeer2>
```

For more information on configuring NTP, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/netmgmt/configuration/guide/nm_basic_sys_manage_ps6350_TSD_Products_Configuration_Guide_Chapter.html#wp1001170

Local Device Traffic Statistics

Local device statistics are the most basic and ubiquitous form of telemetry available. They provide baseline information such as per-interface throughput and bandwidth statistics, enabled features and global per-protocol traffic statistics.

In Cisco IOS, this information is accessed from the command line interface (CLI). The format of a command output, as well as the command itself and its options, vary by platform. It is important to review and understand these differences. The most commonly used commands can be aliased to enable greater operational ease of use.

Per-Interface Statistics

In Cisco IOS, per-interface statistics are available which include throughput (pps) and bandwidth (bps) information. Per-interface statistics can be accessed with the **show interface** command:

```
Router#show interface gigabitEthernet 4/48
GigabitEthernet4/48 is up, line protocol is up (connected)
  Hardware is C6k 1000Mb 802.3, address is 0013.5f21.6c80 (bia 0013.5f21.6c80)
  Description: cr17-3845-1 fe0
  Internet address is 10.139.5.8/31
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mb/s
  input flow-control is off, output flow-control is off
  Clock mode is auto
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:03, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
 Input queue: 0/75/15005/235 (size/max/drops/flushes); Total output drops: 1
 Queueing strategy: fifo
 Output queue :0/40 (size/max)
 5 minute input rate 4751000 bits/sec, 3006 packets/sec
 5 minute output rate 4499000 bits/sec, 2755 packets/sec
 L2 Switched: ucast: 19841909032 pkt, 3347755205145 bytes - mcast: 96885779 pkt, 
5131184435 bytes
 L3 in Switched: ucast: 27282638229 pkt, 5095662463006 bytes - mcast: 94 pkt, 5191 bytes 
mcast
 L3 out Switched: ucast: 43107617667 pkt, 7275264441541 bytes
   47118207406 packets input, 9306459456266 bytes, 0 no buffer
   Received 83653389 broadcasts, 0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 649 overrun, 0 ignored
   0 input packets with dribble condition detected
   43210876182 packets output, 8089398934796 bytes, 0 underruns
   0 output errors, 0 collisions, 0 interface resets
   0 babbles, 0 late collision, 0 deferred
   0 lost carrier, 0 no carrier
   0 output buffer failures, 0 output buffers swapped out
```

For more information on the **show interface** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/interface/command/reference/int_s3g.html

Cisco IOS routers are set by default to use a 5 minute decaying average for interface statistics. Setting the decaying average to one minute provides more granular statistics. The length of time for which data is used to compute load statistics can be changed by using the load-interval interface configuration command.

```
Router(config)# interface <interface-type number>
Router(config)# load-interval 60
```

For more information on the **load-interval interface** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/interface/command/reference/ir_l1.html#wp1014158

The Cisco IOS **pipe** command and its parsing options may also be used to target specific information in the interface output. For example, to quickly view the one minute input and output rates on an interface:

```
Router#show interface <interface-type numer> | include 1 minute
 1 minute input rate 54307000 bits/sec, 17637 packets/sec
 1 minute output rate 119223000 bits/sec, 23936 packets/sec
```

Note High input or output rates over a period of a minute or so can be very helpful in detecting anomalous behavior.

Clearing the interface counters is often necessary to see what is occurring in a particular instance. However, ensure useful information is not being discarded prior to doing so. To clear interface counters:

```
Router#clear counters <interface-type number>
```

Per-Interface IP Feature Information

In Cisco IOS, per-interface feature information provides information about the IP features configured on an interface. In particular, this command is useful to identify the number or name of the ACL being enforced, in order to check the ACL counter hits. Per-interface feature information can be accessed with the **show ip interface** command:

```
Router#show ip interface <interface-type number>
!
Router#show ip interface FastEthernet 2/0
FastEthernet2/0 is up, line protocol is up
  Internet address is 198.133.219.6/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is 110
  Proxy ARP is disabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP Feature Fast switching turbo vector
  IP Feature CEF switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is disabled
  Network address translation is disabled
  BGP Policy Mapping is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect inbound is disabled
  WCCP Redirect exclude is disabled
Router#
```

The **show ip interface** command also provides per-interface uRPF dropped packet statistics. The Cisco IOS **pipe** command and its parsing options can be used to quickly access this information, as shown below.

```
Router#show ip interface <interface-type number> | include 1 verification
!
Router#show ip interface FastEthernet 2/0| include veri
 IP verify source reachable-via ANY
  794407 verification drops
  1874428129 suppressed verification drops
```

For more information on the **show ip interface** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/interface/command/reference/int_s3g.html#wp1205362

Global IP Traffic Statistics

In Cisco IOS, global IP statistics provide a lot of useful information, including per-protocol counts for ICMP, TCP, UDP, and multicast traffic. Global IP traffic statistics can be accessed with the **show ip traffic** command:

```
Router#show ip traffic
IP statistics:
  Rcvd:  4744853 total, 4650886 local destination
         0 format errors, 0 checksum errors, 0 bad hop count
         0 unknown protocol, 0 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 cipso, 0 ump
         0 other
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 fragments, 0 couldn't fragment
  Bcast: 1432237 received, 9 sent
  Mcast: 3156376 received, 3147383 sent
  Sent:  3213086 generated, 284 forwarded
  Drop:  42692 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop
         0 options denied
  Drop:  0 packets with source IP address zero
  Drop:  0 packets with internal loop back IP address
…
ARP statistics:
  Rcvd: 1419832 requests, 4643 replies, 300822 reverse, 0 other
  Sent: 1057 requests, 4897 replies (0 proxy), 0 reverse
```

This command is very useful for general troubleshooting, as well as for detecting anomalies.

The **show ip traffic** command also provides global uRPF dropped packet statistics. The Cisco IOS pipe command and its parsing options may be used to quickly access this information, as shown below.

```
Router#show ip traffic | include RPF
     0 no route, 124780722 unicast RPF, 0 forced drop
```

For more information on the **show ip traffic** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/ipaddr/command/reference/ip1_s2g.html#wp1081111

System Status Information

Memory, CPU and Processes

A basic indication of a potential issue on a network infrastructure device is high CPU.

In Cisco IOS, information about CPU utilization over a 5-second, 1-minute, and 5-minute window is available with the command:

```
Router#show processes cpu
```

The Cisco IOS **pipe** command and its parsing options may be used to exclude information which is not consuming any CPU.

```
Router#show processes cpu | exclude 0.00%__0.00%__0.00%
CPU utilization for five seconds: 38%/26%; one minute: 40%; five minutes: 43%
```

```
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
   5   192962596  13452649      14343  0.00%  0.52%  0.44%   0 Check heaps
  15   4227662201540855414        274  0.65%  0.50%  0.49%   0 ARP Input
  26   2629012683680473726         71  0.24%  0.29%  0.36%   0 Net Background
  50     9564564  11374799        840  0.08%  0.07%  0.08%   0 Compute load avg
  51    15291660    947844      16133  0.00%  0.03%  0.00%   0 Per-minute Jobs
  58    15336356  92241638        166  0.08%  0.02%  0.00%   0 esw_vlan_stat_pr
  67    10760516 506893631         21  0.00%  0.01%  0.00%   0 Spanning Tree
  68  31804659682556402094       1244  7.02%  7.04%  7.75%   0 IP Input
  69    25488912  65260648        390  0.00%  0.03%  0.00%   0 CDP Protocol
  73    16425564  11367610       1444  0.08%  0.02%  0.00%   0 QOS Stats Export
  81    12460616   1020497      12210  0.00%  0.02%  0.00%   0 Adj Manager
  82   442430400  87286325       5068  0.65%  0.73%  0.74%   0 CEF process
  83    68812944  11509863       5978  0.00%  0.09%  0.11%   0 IPC LC Message H
  95    54354632  98373054        552  0.16%  0.12%  0.13%   0 DHCPD Receive
  96    61891604  58317134       1061  1.47%  0.00%  4.43%   0 Feature Manager
```

High CPU utilization values for the IP Input process is a good indicator that traffic ingressing or egressing the device is contributing meaningfully to CPU load. The amount of process-driven traffic versus interrupt-driven traffic is also important.

Understanding the network devices deployed in your network and their normal status is key to establishing a baseline, from which anomalies may be detected.

For more information on the **show proc cpu** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g09.html#wp1042641

Memory Threshold Notifications by Syslog

Cisco IOS offers the ability to send a notification upon memory thresholds being exceeded. A syslog message is sent when memory utilization falls below a configurable low watermark and when free memory once again reaches the configured threshold. Low watermarks can be defined for both processor and input/output (I/O) memory with the following commands:

```
Router(config)#memory free low-watermark processor <kilobytes threshold>
Router(config)#memory free low-watermark io <kilobytes threshold>
```

Once the thresholds are configured, the router will issue a notification every time the available free memory falls below the specified threshold, and every time the available free memory rises to 5 percent above the specified threshold.

Example output when available free processor memory less than the specified threshold:

```
000029: *Aug 12 22:31:19.559: %SYS-4-FREEMEMLOW: Free Memory has dropped below 2000k
Pool: Processor  Free: 66814056  freemem_lwm: 204800000

Example output when available free processor memory recovered to more than the specified
threshold
000032: *Aug 12 22:33:29.411: %SYS-5-FREEMEMRECOVER: Free Memory has recovered 2000k
Pool: Processor  Free: 66813960  freemem_lwm: 0
```

For more information on the Memory Threshold Notifications feature, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fs_memnt.html

Reserving Memory for Critical Notifications

Cisco IOS offers the ability to preserve critical system logging when a device is overloaded and system resources are low. This feature reserves a region of memory on the device which is only available for critical system logging. Critical system logging memory reservation is enabled with the following command:

```
Router(config)# memory reserve critical <kilobytes>
```

Note The amount of memory reserved for critical notifications cannot exceed 25 percent of total available memory.

For more information on the **memory reserve critical** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fs_memnt.html#wp1057054

CPU Threshold SNMP Trap Notification

Cisco IOS offers the ability to send a notification upon CPU thresholds being exceeded. An SNMP trap can be sent when CPU utilization exceeds a configurable high-water mark, and when CPU utilization falls below a configurable low-water mark, within a configurable window.

Sudden increases in CPU load on routers and switches often indicate that some event is taking place, however high CPU is not always an indicator of malicious activity. Therefore, analysis and correlation of other event information is highly recommended.

CPU threshold notification can be defined for:

- Total CPU utilization
- CPU process utilization
- CPU interrupt utilization

CPU Thresholding Notification can be configured with the **process cpu threshold** command:

```
Router(config)# process cpu threshold type {total | process | interrupt} rising
<percentage> interval <seconds> [falling <percentage> interval <seconds>]
```

For example, to send an SNMP trap upon total CPU utilization exceeded 80% for more than 5 seconds and being below 20% for more than 5 seconds:

```
Router(config)# snmp-server enable traps cpu threshold
Router(config)# snmp-server host 172.26.150.206 traps public cpu
Router(config)# process cpu threshold type total rising 80 interval 5 falling 20 interval 5
```

It is also a good practice to set the process entry limit and the size of the history table for CPU utilization statistics with the **process cpu statistics** command:

```
Router(config)# process cpu statistics limit entry-percentage number [size seconds]
```

The following example shows how to set an entry limit at 40 percent and a size of 300 seconds:

```
Router(config)# process cpu statistics limit entry-percentage 40 size 300
```

For more information on the **process cpu threshold type** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_m1.html#wp1011517

For more information on the **process cpu statstics limit entry-percentage** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_m1.html#wp1011684

MAC Address Table Status

Information on which MAC addresses are currently connected to which port can be useful for traceback upon anomalous behavior being detected.

Cisco IOS offers the ability to view the status of the MAC address table on Catalyst Switches using the following command:

```
Router#show mac-address-table
```

The output includes information on currently stored MAC addresses, if they are static or dynamically learnt, their age, and associated VLAN and interface.

This command can be used to trace a specific MAC address, as shown below for the MAC address 0100.5e00.0128 on a Catalyst 6500 with a Supervisor Engine 720:

```
Router# show mac-address-table address 0100.5e00.0128
Legend: * - primary entry
        age - seconds since last seen
        n/a - not available

  vlan   mac address     type    learn     age              ports
------+----------------+--------+-----+----------+-------------------------
Supervisor:
*   44  0100.5e00.0128    static  Yes          -   Fa6/44,Router
*    1  0100.5e00.0128    static  Yes          -   Router
Module 9:
*   44  0100.5e00.0128    static  Yes          -   Fa6/44,Router
*    1  0100.5e00.0128    static  Yes          -   Router
```

The number of MAC addresses currently stored in the MAC address table and the amount of space remaining can be viewed with the `mac-address-table count` command. An sample output for a particular slot on a Catalyst 6500 is shown below.

```
Router# show mac-address-table count slot 1
MAC Entries on slot 1 :
Dynamic Address Count:                4
Static Address (User-defined) Count:  25
Total MAC Addresses In Use:           29
Total MAC Addresses Available:        131072
```

MAC address table entries will age out according to the configured aging time. Dynamically learnt MAC addresses can be cleared with the following command:

```
clear mac-address-table dynamic
```

For more information on the **show mac-address-table** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/switch/command/reference/swi_s3.html#wp1074753

Open Ports and Sockets

The ports and sockets open on a device should be reviewed to ensure that unused or unneccesary ports and sockets are disabled. In Cisco IOS, open ports and sockets can be viewed with the **show control-plane host open-ports** command:

```
Router#show control-plane host open-ports
Active internet connections (servers and established)
Prot        Local Address      Foreign Address                  Service    State
 tcp                 *:22                  *:0               SSH-Server   LISTEN
 tcp                 *:23                  *:0                   Telnet   LISTEN
 tcp              *:63771    172.26.150.206:49         IOS host service ESTABLIS
 udp                 *:49     172.26.150.206:0           TACACS service   LISTEN
 udp                 *:67                  *:0            DHCPD Receive   LISTEN

Router#
```

The **show control-plane host open-ports** command was introduced in 12.3(4)T. For earlier versions of Cisco IOS follow the steps below:

To check open UDP ports use the **show ip sockets** command:

```
Router#show ip sockets
```

A sample output is shown below:

```
Router#show ip sockets
Proto    Remote      Port      Local       Port  In Out Stat TTY OutputIF
 17 0.0.0.0             0 198.133.219.6      67   0   0 2211   0
 88   --listen--          --any--           100   0   0    0   0
 17 172.26.150.206     49 172.26.159.165     49   0   0   21   0
 17   --listen--          --any--           161   0   0    1   0
 17   --listen--          --any--           162   0   0   11   0
 17   --listen--          --any--         56443   0   0    1   0
 17 172.26.150.206    514 172.26.159.165  55759   0   0  210   0
Router#
```

In earlier Cisco IOS versions, open TCP ports can be viewed in two steps with the following commands:

```
Router#show tcp brief all
Router#show tcp tcb
```

Use the **show tcp brief all** command to see the IP source and destination IP addresses and state of TCP sessions. This command also provides the transmission control block (TCB), an internal identifier used by the router/switch to identify the connection. The TCB values are then used to identify the ports associated with the connections.

```
Router#show tcp brief all
TCB       Local Address               Foreign Address             (state)
661BB46C  172.26.159.165.49128        172.26.150.206.49           ESTAB
6612A398  198.133.219.6.179           198.133.219.10.11003        ESTAB
20465FC8  172.26.159.165.22           172.26.159.164.15774        ESTAB
50711308  198.133.219.6.16422         198.133.219.5.179           ESTAB
661B9248  172.26.159.165.19110        172.26.150.206.49           CLOSEWAIT
6612ACC4  *.179                       198.133.219.5.*             LISTEN
661294C0  *.179                       198.133.219.10.*            LISTEN
Router#
```

Use the **show tcp tcb** command to identify the source and destinations sockets for a given TCP session:

```
Router#show tcp tcb 20465FC8
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 172.26.159.165, Local port: 22
Foreign host: 172.26.159.164, Foreign port: 15774
```

```
Connection tableid (VRF): 0
…
```

CDP Best Common Practices

Cisco Discovery Protocol (CDP) is a device discovery protocol that runs over Layer 2 and allows Cisco devices to exchange information with each another, including adjacency information. This can be useful for traceback in the case of a network incident.

CDP information is transferred in clear text and communication is unauthenticated. Consequently, CDP is vulnerable to abuse, including sniffing to learn sensitive information about a network infrastructure device, such as IP address, software version, router model, and network topology.

CDP is, however, very useful to OPSEC (operational security) personnel for traceback during an incident, enabling hop-by-hop investigation.

The best common practice for CDP is currently:

- Enable CDP on point-to-point infrastructure inks
- Disable CDP on edge devices or interfaces where is it not required and where such as service may represent a risk, including:
  - LAN access edge
  - Internet transit edge
  - Extranet edge
  - Any public-facing interfaces

It should be noted that CDP is required for some Cisco applications, products and features, such as Cisco IP Telephony for network management.

In the rare case CDP is not used for troubleshooting or security operations, the service can be globally disabled using the **no cdp run** command:

```
Router(config)#no cdp run
```

The CDP service may be disabled on a per- interface basis using the **no cdp enable** interface command:

```
Router(config)# interface <interface-type number>
Router(config-if)# no cdp enable
```

For more information on the **cdp run and cdp enable** command, refer to the following URLs:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g01.html#wp1032125

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g01.html#wp1031940

CDP Neighbor Information

To view the detailed information CDP discovers about neighboring devices, use the following command:

```
Router#show cdp neighbors
```

A sample output is shown below:

```
cr18-7301-1#sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater
```

```
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
cr17-2821-1      Gig 0/2            154        R S I      2821      Gig 0/0
cr18-6500-2.cisco.com
                 Gig 0/1            150        R S I      WS-C6506  Gig 2/1
cr18-6500-1.cisco.com
                 Gig 0/0            123        R S I      WS-C6506-EGig 2/1
cr18-7301-1#
```

For more information on the **show cdp neighbors** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/cfr_1g07.html#wp1032872

Syslog

Syslog is a UDP-based logging facility enabling detailed messages to be sent from a device to a syslog server. Syslog packets are reactively sent based on the occurrence of specific events on a device and provide invaluable operational information, including system status, traffic statistics and device access information.

Syslog data is critical to recording day-to-day event and debugging information, as well as notifying operational staff of critical system alerts. Cross-network data aggregation to a central syslog server enables detailed and behavioral analysis of the data which is key to incident handling and attack forensics, as well as general network visibility and routine troubleshooting.

Due to is invaluable role in cross-network data aggregation, syslog should be enabled not just on routers, firewalls and switches, but also on hosts and applications, such as DNS, TACACS+ and RADIUS.

Syslog Best Common Practices

One of the challenges with syslog is that the amount of syslog data can create significant load on both the device sending the data and the syslog server. Cisco IOS, by default, sets syslog logging to level 6 (informational), which can generate a lot of output and can possibly impact the device CPU. Consequently, it is critical to ensure that the following considerations are taken into account:

- Log syslog messages to a central server
- Ensure the syslog server has adequate storage and processing capacity
- Syslog rate-limiting is recommended, where available
- Selectively enabled more detailed logging on critical systems or systems exposed to external users
- Use facility numbers to enable the syslog output to be more easily organized on the syslog server, by default, Cisco routers export syslog as facility ‘local7’
- Do not log to the console on Cisco IOS devices
- Define the source IP address to be used for syslog messages. This will typically be a loopback or an OOB interface to ensure consistency.
- Ensure timestamps are enabled per the guidelines in the section Timestamps and NTP
- Ensure syslog messages are stored in a searchable database
- The syslog server should be properly hardened and cryptographic techniques should be considered to protect the logs

- Syslog communication is not authenticated and data is not encrypted. Consequently, syslog packets are susceptible to sniffing. If the confidentiality, integrity and reliability of syslog messages is a concern, an IPSec tunnel/connection from devices to the syslog servers may be considered.

Syslog to a Central Server

In Cisco IOS, syslog messages can be sent to a central server by defining the source IP address to be used, the syslog server(s) to which to send the messages, and the logging trap level.

```
Router(config)#logging source-interface <interface>
Router(config)#logging host <syslogserver>
Router(config)#logging trap <level>
```

Table 5-2 Error Message Severity Levels, Equivalent Text, and Descriptions

Numeric Severity Level	Equivalent Word	Description
0	**emergencies**	System unusable
1	**alerts**	Immediate action needed
2	**critical**	Critical conditions
3	**errors**	Error conditions
4	**warnings**	Warning conditions
5	**notifications**	Normal but significant condition
6	**informational**	Informational messages only
7	**debugging**	Debugging messages

Cisco Security Monitoring, Analysis, and Response System (MARS) can be used to collect, analyze and correlate event information generated from a diverse se of network devices and host applications, from Cisco and other vendors. MARS security monitoring can be used in combination of other forms of telemetry, helping reduce false positives and improving threat identification and mitigation responses.

For more information on the **logging source-interface** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_07.html#wp1015076

For more information on the **logging host** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_07.html#wp1020426

For more information on the **logging trap** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_07.html#wp1015177

Syslog Named Facilities

The use of multiple named syslog facilities can be used to logically separate and store syslog messages. Each facility name can be stored and processed in different directory location on the syslog server. For instance, a logging facility directory structure may be implemented based on the role of a device, such as core router, internal edge device, external edge device, etc. Each individual facility may then be processed, reviewed and acted upon according to operational needs.

In Cisco IOS, syslog named facilities are configured using the following command:

```
Router(config)#logging facility <facility>
```

For more information on the **logging facility** command, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_07.html#wp1012726

Syslog Rate-Limiting

Syslog rate-limiting, if available, is recommended to ensure that syslog messages do not impact the CPU of either the sending device or the syslog server. As the name indicates, syslog rate-limiting limits the rate of messages logged per second.

In Cisco IOS versions supporting syslog rate-limiting, syslog is limited to 10 messages per second by default. This default behavior can be changed with the `logging rate-limit` command. The `logging rate-limit` command allows you set a different rate limit for all severity levels, or for severity levels above an specified value.

In the following example, messages of any severity level are limited to 5 per second:

```
Router(config)#logging rate-limit 5
```

In this example, messages at level 3 and above are limited to one message per second, while syslog messages at levels 0-2 (emergencies, alerts and critical) are not rate-limited.

```
Router(config)#logging rate-limit 1 except 2
```

For more information on the **logging rate-limit** command and syslog facilities, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/netmgmt/command/reference/nm_07.html#wp1014866

Common Syslog Servers

Some syslog servers in common use include:

- Simple Event Correlator (SEC)

  A powerful open source tool which enables parsing and correlation of syslog output. It requires some scripting skills to use but is quite useful.

- Cisco CS-MARS

  Takes syslog input from routers, switches, firewalls, IDS, VPN concentrators and combines it with other forms of telemetry in order to provide anomaly-detection and event correlation. It can be configured to accept syslog output from many different types of network devices, servers, etc.

- Sawmill

  A commercial Web-based tool used to analyze many different kinds of syslog output and render HTML reports and graphs.

  Kiwi Syslog Daemon: http://www.kiwisyslog.com/kiwi-syslog-daemon-overview/

Open source databases such as MySQL and PostGres are often used to store syslog output for post-processing and searching.

SNMP

Simple Network Management Protocol (SNMP) is a popular network management protocol that provides a range of information useful for network telemetry.

A general overview of SNMP, including information on the different versions and their associated security mechanisms, as well as general guidelines, can be found in the section SNMP Access.

Common SNMP Servers

Some tools and management platforms in common use which leverage SNMP telemetry include:

- Open source tools including NET-SNMP, MRTG, Cricket, RRDTool, Nagios
- Cisco MARS and Arbor PeakFlow DoS which combine SNMP polling of interface speeds with NetFlow telemetry in order to perform traffic profiling and anomaly-detection
- NMS platforms such as HP OpenView and Nagios
  Often support alerting based upon low-water-marks as well as high-water-marks

ACL Logging

ACL logging can be used to log basic information related to:

- Successful access attempts from authorised communicators
- Failed access attempts from non-authorised communicators

The use of an ACL with the ‘log’ keyword provides information on the IP addresses and port numbers associated with a packet. The use of an extended ACL with the ‘log-input’ keyword extends this information to also include the input interface and source MAC or VC. Consequently, the ‘log-input’ keyword should be used whenever extended ACLs are available.

It is recommended that ACL logging is only enabled on an ACL that will not normally match a very large number of packets. This ensures the log remains manageable and does not overflow the restricted log file size. ACL log messages are rate-limited and so any anomalies in this expected behavior will not severely impact the device availability.

- Router Access Control List (ACL) logging is very common; however, it can generate a lot of information in a short span of time, and have a negative impact on performance. ACL logging can be rate-limited, but there are other forms of telemetry such as NetFlow which can provide more information about network traffic
- The Cisco ASA firewall appliance and the Firewall Services Module (FWSM) for the 6500/7600 also generate ACL logs; the performance impact of ACL logging on these devices is much less than on routers

Accounting

Accounting is a critical element of network telemetry and is covered in Infrastructure Device Management Access Logging, page 2-25.

Configuration Change Notification and Logging

Cisco IOS offers a Configuration Change Notification and Logging (Config Log Archive) feature which tracks commands executed in configuration mode through the CLI or HTTP. This feature is covered in Infrastructure Device Management Access Logging, page 2-25.

Packet Capture

Packet capture is generally undertaken after a macro-level indication of an anomaly, for instance via SNMP or syslog, in order to enable more detailed analysis.

General guidelines include:

- Packet capture should take place at key points in the topology such as distribution gateways, IDC switch meshes, desktop access switch meshes, and in some cases, the core.
- It is important to be as specific as possible when capturing packets; at high rates of speed, the amount of information can be overwhelming.
- It is extremely important to ensure that traffic is captured bidirectionally, or, if this is not possible, to identify the unidirectional nature of the capture and take this into account when analyzing the captured traffic.
- Conversely, it is important to avoid capturing duplicate traffic, especially in complex topologies.

SPAN/RSPAN

Switchport Analysis and Remote Switchport Analysis (SPAN/RSPAN) are Cisco IOS features that enable packets to be passed to traffic analysis systems.

SPAN sessions allow you to monitor traffic on one or more ports, or one or more VLANs, and send the monitored traffic to one or more destination ports.

SPAN/RSPAN does not a have measurable performance impact on the network device and does not affect the switching of traffic on source ports or VLANs. SPAN sends a copy of the packets received or transmitted by the source ports and VLANs to the destination port. The destination port must be dedicated for use by SPAN.

For more information on SPAN on Catalyst 6500 Series Switches, see the configuration guide at the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SXF/native/configuration/guide/span.html

Copy/Capture VLAN ACLs

The Cisco IOS VLAN ACL (VACL) feature may be leveraged to enable packets to be passed to traffic analysis systems. The VACL action clause “forward capture” is used to allow packets to continue to flow but to copy them to ports configured as “capture ports”; a port configured to capture VACL-filtered traffic.

An example of how to define and apply a copy/capture VACL access map to forward and capture all IP traffic matching net_10 is shown below.

```
Router(config)# vlan access-map capture1 10
Router(config-access-map)# match ip address net_10
Router(config-access-map)# action forward capture
Router(config-access-map)# exit
Router(config)# vlan filter capture1 vlan-list 2, 4-6
```

An interface can be configured as a capture port with the following command:

```
Router(config)# interface interface
Router(config-if)# switchport capture
```

The capture action sets the capture bit for the forwarded packets so that ports with the capture function enabled can receive the packets. Only forwarded packets can be captured.

A capture port supports only egress traffic. No traffic can enter the switch through a capture port. VACLs do not a have measurable performance impact on the network device.

For more information on VACLs on Catalyst 6500 Series Switches, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SXF/native/configuration/guide/vacl.html

The Cisco NAM-2 captures packets via SPAN/RSPAN or copy/capture VACLs on the 6500/7600. It can perform basic on-board analysis but captures are typically saved and downloaded for use by a dedicated traffic analysis device, such as Ethereal or Network General Sniffer.

General Network Telemetry Indicators

- CPU
  - Spikes in CPU load on network infrastructure devices are often an indication that an event is taking place.
  - High CPU is not always an indicator of malicious activity. Trending is important.
  - Correlating CPU utilization with other information such as network traffic statistics, routing-table changes, etc. is very useful.
  - A baseline of CPU utilization over time is a good idea from a network management standpoint, and also allows operational staff to determine if further investigation is warranted.
- Traffic Rates
  - Excessive bandwidth (bps) and/or throughput (pps) can be an indicator of undesirable traffic.
  - DoS attacks and DoS-like worms can cause high amounts of traffic.
  - High amounts of traffic are not always indicative of problems, since it is dependent on the situation at any particular time. This is another reason why a telemetry baseline is important.
  - Most Network Management Systems (NMS) can alert on high bps and/or high pps. It can be just as important to be notified of a drop in link speed (or number of routes, etc.) as it is to learn of an increase.
- Link Flaps
  - Link-flaps can indicate that something is amiss.
  - Link flaps are often a sign of miss-configuration, backhole and other similar incidents. However, link flaps can also result from malicious activity, such as a DoS attack.
  - Routers and switches can be configured to notify monitoring systems when a link flap occurs.
  - Link flaps are not always indicative of problems, since it is dependent on the situation at any particular time. This is another reason why a telemetry baseline is important.

CHAPTER 6

Network Policy Enforcement

Baseline network policy enforcement is primarily concerned with ensuring that traffic entering a network conforms to the network policy, including the IP address range and traffic types. Anomalous packets should be discarded as close to the edge of the network as possible, thereby minimizing the risk of exposure.

This section highlights the key steps to implementing baseline network policy enforcement, including:

- Access Edge Filtering
- IP Spoofing Protection

CSF Methodology Assessment

The results of applying the CSF methodology for baseline network policy enforcement are presented in Table 6-1 and highlight the technologies and features identified for enforcing baseline network policy enforcement and which are integrated in Network Security Baseline.

Total Visibility

Table 6-1 Network Policy Enforcement—Total Visibility

Identify	Monitor	Correlate
	• Logging – Syslog – SNMP	

Complete Control

Table 6-2 Network Policy Enforcement—Complete Control

Harden	Isolate	Enforcement
	• ACL	• Access Edge Filtering – iACLs • IP Spoofing Protection – uRPF – IP Source Guard – DHCP/ARP enforcement

Access Edge Filtering

The Network Security Baseline is focused on securing the network infrastructure itself, the control and management planes. Access edge filtering in this context is implemented to enforce policy on what traffic is permitted to be directed towards the network infrastructure devices themselves.

In Cisco IOS, access edge filtering for control and the data planes is achieved using ACLs developed to protect the infrastructure. These are referred to as infrastructure protection ACLs (iACLs).

Detailed information on the role, development and configuration of iACLs is provided in Infrastructure Protection ACLs (iACLs), page 8-9.

IP Spoofing Protection

Spoofing protection involves discarding traffic that has an invalid source address. Network Security Baseline includes source IP spoofing protection based on BCP38/RFC 2827 ingress traffic filtering.

Packets with spoofed source IP addresses represent a security risk as they are often used to conduct an attack, in order to evade traceability and bypass access controls. They may also be used to direct an attack at a spoofed source, something known as a “reflection attack”.

Table 6-3 IP Spoofing Protection

Type of Spoofing	Related Attacks	Possible Countermeasures
IP spoofing	ICMP unreachable storm	• ACLs • uRPF • IP Source Guard • DHCP Secured IP Address Assignment and DHCP Authorized ARP
	ISYN flood	
	SYN flood	
	Spoof trusted IP addresses to leverage trust relationship	
	DDoS	

Spoofed traffic with an invalid source IP address may include traffic from a:

- RFC1918, DSUA or non-allocated IP address range
- Valid IP network address range but not originating from the associated legitimate network

Implementing BCP38/RFC 2827 ingress traffic filtering to address source IP address spoofing renders the use of invalid source IP addresses ineffective, forcing attacks to be initiated from valid, reachable IP addresses. This is beneficial since it enables greater success in tracing the originator of an attack.

IP spoofing protection offers the following key benefits:

- Enables improved, clearer analysis of network telemetry data
- Increases traceback success
- Improves the traceability of the source of malicious behavior
- Eliminates bogon-sourced traffic from the peering edge
- Aides the effectiveness of iACLs

Cisco offers the following techniques for BCP38 ingress traffic filtering:

- Access Control Lists (ACLs)

  ACLs are the traditional technique for filtering forged source IP addresses. However, ACLs are not dynamic in nature, requiring manual configuration changes, and may have an impact on the performance of a device. It is thus recommended that ACLs are used only in a limited manner, as a complement to uRPF, for strict, static policies, such as filtering RFC 1918, DSUA and non-allocated IP addresses. They may also be used to complement uRPF loose mode for source IP address spoofing protection when uRPF strict mode is not possible. ACLs are covered in Appendix 4, "Device Resiliency and Survivability."

- uRPF

  uRPF offers a dynamic technique for enabling BCP38/RFC 2827 ingress traffic filtering, discarding packets with invalid source IP addresses based on a reverse-path look-up. Its dynamic nature provides the key advantages of offering minimal operational overhead and a scalable, timely enforcement technique. In addition, uRPF generally introduces minimal performance impact to a device on which it is enabled. uRPF is typically deployed as an edge technology in order to be most effective, minimizing the valid IP address space range and enforcing the discard of anomalous packets as close to their origin as possible.

- IP Source Guard

  IP Source Guard is used in switched environments to prohibit the use of forged MAC and source IP addresses. This feature is deployed on Layer 2 switching devices and is primarily designed for DHCP segments. Hosts with static address may also be supported, though additional operational complexity is introduced by doing so.

- DHCP Secured IP Address Assignment and DHCP Authorized ARP

  These Cisco IOS features are available on routers supported on the T-train and offer similar functionality in a routing environment as IP Source Guard in a switching environment. They are used in routed environments where the local router is also the local DHCP server to prohibit the use of forged MAC and source IP addresses

  http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ftdsiaa.html
  http://www.cisco.com/en/US/docs/ios/12_3t/12_3t4/feature/guide/gtautarp.html

Network Security Baseline includes ACLs and uRPF to address IP spoofing protection, as these are the simplest techniques to implement and form a baseline upon which additional spoofing protection techniques may be deployed.

ACLs are covered in Chapter 4, "Device Resiliency and Survivability."

Unicast Reverse Path Forwarding (uRPF)

uRPF offers a dynamic technique for enabling BCP38/RFC 2827 ingress traffic filtering, discarding packets with invalid source IP addresses based on a reverse-path look-up. Its dynamic nature provides the following key advantages:

- Minimal operational overhead
- Scalable, timely enforcement technique
- Generally introduces minimal performance impact to a device on which it is enabled.

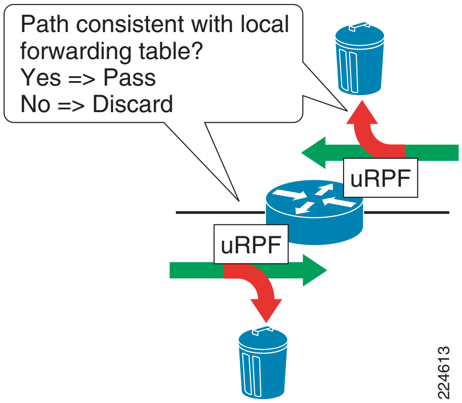
uRPF is thus a highly attractive alternative to traditional ACLs, which typically demand significant management overhead and have a greater impact on device performance.

uRPF is typically deployed as an edge technology in order to be most effective, minimizing the valid IP address space range and enforcing the discard of anomalous packets as close to their origin as possible.

The key function of uRPF is to verify that the path of an incoming packet is consistent with the local packet forwarding information. This is achieved by performing a reverse path look-up (hence the feature's name) using the source IP address of an incoming packet in order to determine the current path (adjacency) to that IP address. The validity of this path determines whether uRPF will pass or drop the packet.

If the path is valid, the packet will be passed. If the path is not valid, the packet will be silently discarded (unless an ACL exemption is configured). See Figure 6-1.

Figure 6-1 Unicast Reverse Path Forwarding (uRPF)



Once enabled on an interface, uRPF checks all IP packets (IPv4 and IPv6) on the input path of that interface. The key edge locations and the specific objectives of uRPF in each of these locations are shown Table 6-4.

Table 6-4 uRPF Key Edge Locations and Objectives

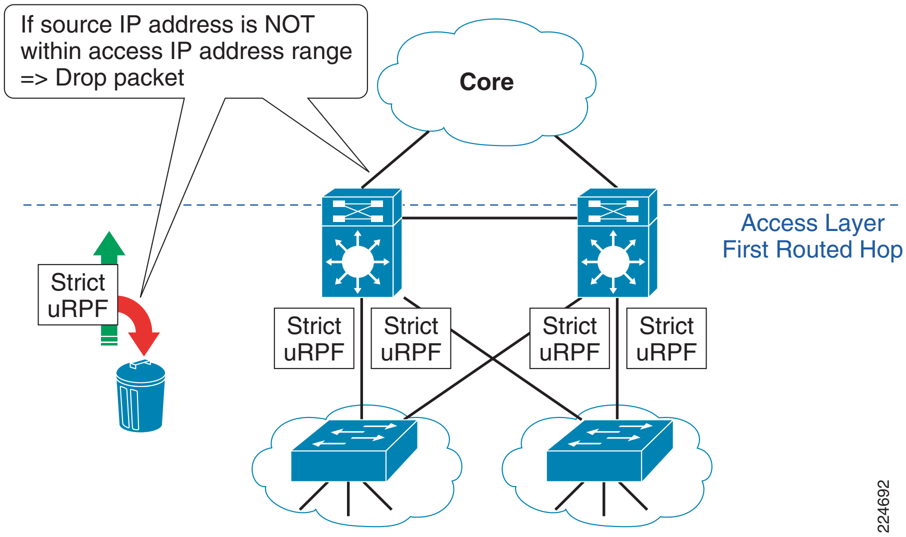
Network Edge Location	Specific uRPF Objectives
Access Layer Edge First Routed Hop[1]	• Discard incoming packets on the first routed hop of the access layer edge that do NOT have a source IP address within the access network IP range
Enterprise Internet Edge	• Discard incoming packets on the internal interface that do NOT have a source IP address within the internal network IP range • Discard incoming packets on the external Internet interface which have a source IP address within the internal network IP range

[1] The term 'Access Layer Edge First Routed Hop' is used to refer to the first Layer 3 routed hop that is encountered in the access layer of a network. In an enterprise deployment, this is often referred to as the 'Distribution Layer Edge' and may be a router or a Layer2/Layer 3 switch

Access Layer First Routed Hop

In this location, ingress traffic filtering for source IP spoofing protection is applied at the first Layer 2-Layer 3 routed boundary of the access layer network (see Figure 6-2).

Figure 6-2 Access Layer First Routed Hop


If source IP address is NOT within access IP address range => Drop packet
Core
Access Layer First Routed Hop
Strict uRPF
Strict uRPF
Strict uRPF
Strict uRPF
Strict uRPF
224692

The key objective is:

Discard incoming packets on the first routed hop of the access layer edge that do NOT have a source IP address within the assigned access network IP range.

As illustrated in Figure 6-2, the general deployment approach is:

Enable uRPF strict mode on all first routed hop interfaces facing the Layer 2 access network downstream devices.

Deployment Considerations

In certain access layer edge scenarios, uRPF strict mode may either not be possible or may require some additional design work. Common scenarios requiring additional consideration include:

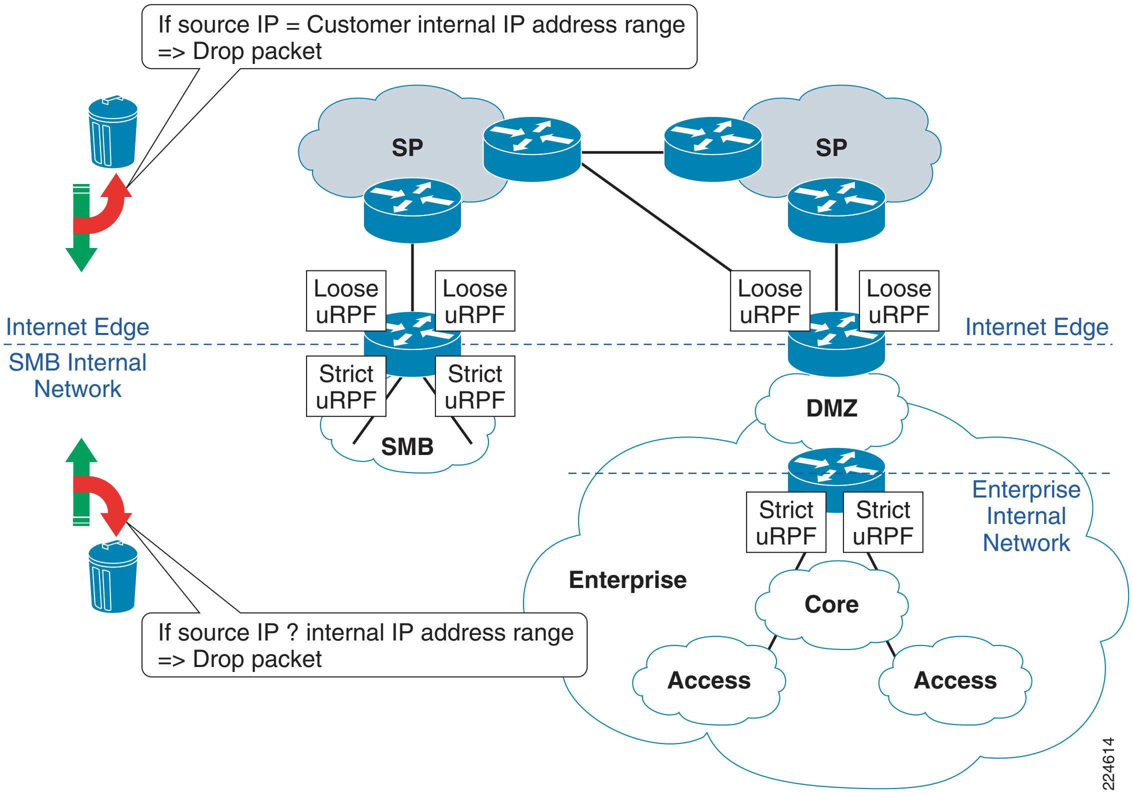
- Access layer topologies with downstream Layer 2 devices which are interconnected for redundancy, thereby creating the possibility of multiple paths to a particular IP prefix.
  This scenario typically creates a challenge with DHCP assignment and management applications.
- Dual-homed hosts which may be configured to send traffic to either gateway

IP Source Guard may be used to supplement uRPF but this is not included in Network Security Baseline.

Enterprise Internet Edge

In this location, ingress traffic filtering for source IP spoofing protection is applied on the SMB or Enterprise's Internet edge devices (see Figure 6-3).

Figure 6-3 Enterprise Internet Edge



The key objectives being:

- Discard incoming packets on the internal interface that do NOT have a source IP address within the internal network IP range.
- Discard incoming packets on the external Internet interface which have a source IP address within the internal network IP range.

As illustrated in Figure 6-3, the general deployment approach is:

- Enable uRPF strict mode on the routed interfaces of first-hop devices facing the internal network
- Enable uRPF loose mode on external routed interfaces facing the Internet, along with source- based ACLs (since uRPF loose mode alone will not meet the design objective)

Deployment Considerations

- uRPF strict mode is only possible on the internal network edge interfaces if all paths to any internal IP prefix are of all equal cost, thereby allowing all valid paths to be present in the FIB. If this is not the case, as may be experienced in networks with multiple Internet connections from geographically disperse sites, uRPF loose mode must be deployed as an alternative.
- On networks with just a single external Internet connection, it may be possible to deploy uRPF strict mode. Review the FIB to verify.
- uRPF strict mode is typically not possible on the external Internet edge for networks which are multi-homed to one or more service providers, since multiple valid paths may exist to an IP prefix. The CEF FIB will typically only contain a subset of these valid paths, due to its 'single best path' selection algorithm, and thus uRPF strict mode may, inadvertently, drop valid packets.
- uRPF loose mode provides only limited source IP spoofing protection, since any path on the device will be valid. Consequently, only packets with source IP addresses not present in the FIB will be dropped.
- Since uRPF loose mode provides only limited source IP spoofing protection, source-based ACLs are required on the external Internet edge in order to supplement uRPF loose mode in achieving our objective of dropping incoming packets with a source IP address matching the internal network IP range.
- uRPF loose mode is extremely valuable in this location as a key enabler for the deployment of source IP based black hole and SRTBH rapid reaction attack tools.
- If the FIB contains a path to valid IP prefixes via a default route, the 'allow-default' uRPF option may be required to ensure valid packets are not inadvertently dropped. Note, however, that this may significantly reduce the effectiveness of uRPF in providing source IP spoofing protection.
- On external Internet interfaces running BGP and multi-homed to a single AS, it may be possible to deploy uRPF strict mode by using BGP parameters to force the FIB to be populated with all valid paths.
- A static route to null0 for bogon IP addresses may also be inserted into the routing table to enable uRPF to drop traffic from these invalid source IP addresses, thereby simplifying ACLs.

For more information on configuring uRPF, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_cfg_unicast_rpf_ps6350_TSD_Products_Configuration_Guide_Chapter.html

CHAPTER 7

Switching Infrastructure

Baseline switching security is concerned with ensuring the availability of the Layer 2 switching network. This section highlights the key steps to securing and preserving the switching infrastructure, including:

- Restrict Broadcast Domains
- Spanning Tree Protocol (STP) Security
- VLAN Best Common Practices

CSF Methodology Assessment

The results of applying the CSF methodology for baseline switching security are presented in Table 7-1 and Table 7-2. The tables highlight the technologies and features identified for baseline switching security and which are integrated in Network Security Baseline.

Total Visibility

Table 7-1 Switching Infrastructure—Total Visibility

Identify	Monitor	Correlate
	• Logging – Syslog – SNMP	

Complete Control

Table 7-2 Switching Infrastructure—Complete Control

Harden	Isolate	Enforce
	• Restrict Broadcast Domains • VLAN • L3 hierarchical design	• STP Security – Disable dynamic trunking – PVST – BPDU Guard – Root guard • VLAN Best Common Practices

Restrict Broadcast Domains

By definition, LAN switches are responsible for forwarding unknown frames, multicast frames and broadcast frames throughout the LAN segment, forming a broadcast domain. While broadcast domains facilitate Layer 2 connectivity between systems on a LAN segment, designing networks with unnecessarily large broadcast domains has potential drawbacks.

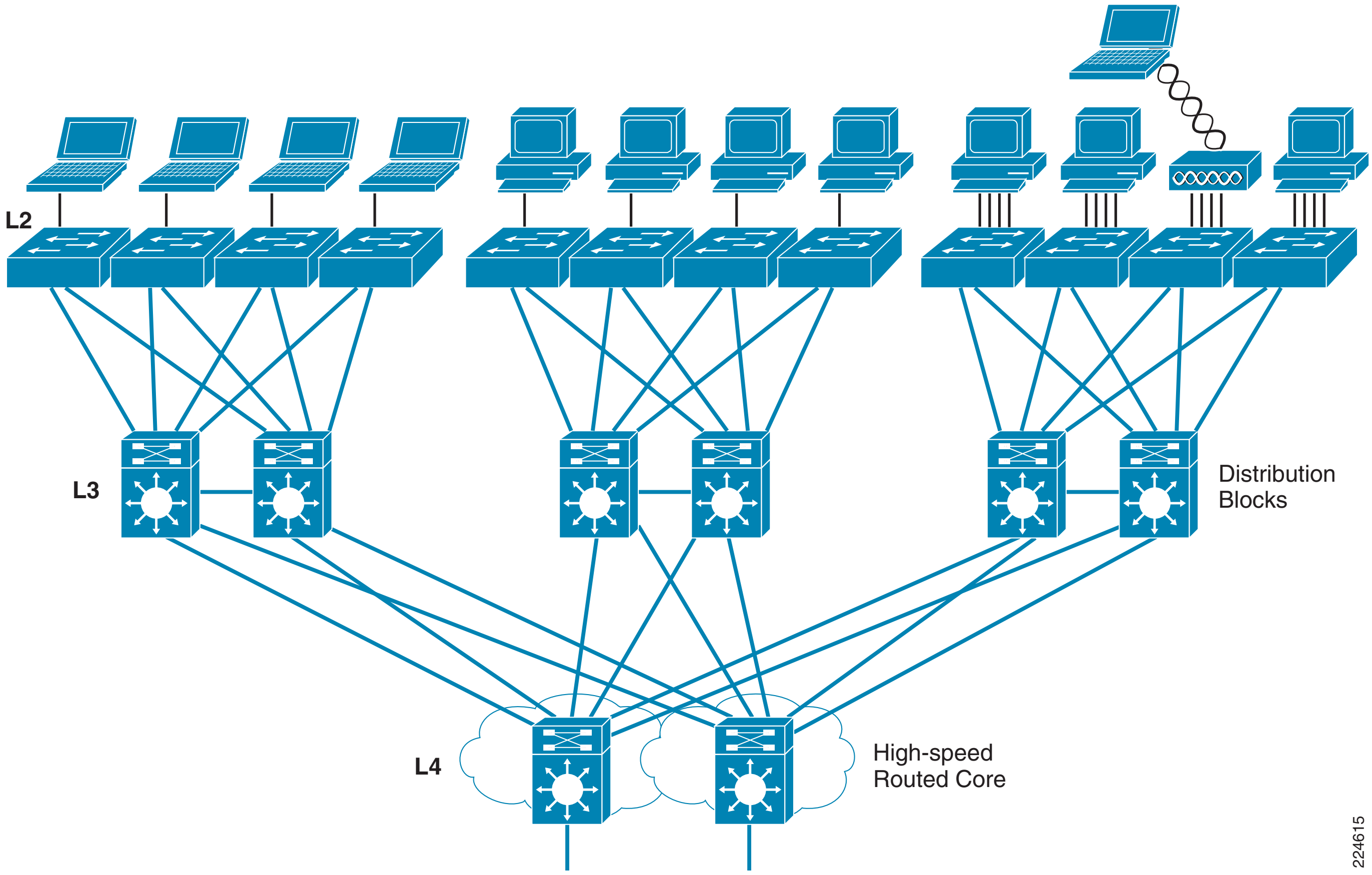
First, in large networks, the flooding of unknown, multicast and broadcast frames may degrade performance, even to the point of breaking connectivity. In addition, a broadcast domain defines a failure domain, whereby typically all systems and switches on the same LAN segment suffer during a failure. Therefore, the larger the broadcast domain, the bigger the impact of a failure. Finally, larger broadcast domains increase the chances of security incidents.

To avoid the challenges described above, it is a good practice to segment broadcast domains into multiple IP subnets or VLANs using a hierarchical design.The use of hierarchical design principles provides the foundation for implementing scalable and reliable LANs.

Figure 7-1 shows a recommended hierarchical design. This design uses a building block approach leveraging a high-speed routed core network layer to which are attached multiple independent distribution blocks. The distribution blocks comprise two layers of switches: the actual distribution nodes that act as aggregators, and wiring closet access switches. The hierarchical design segregates the functions of the network into these separate building blocks to provide for availability, flexibility, scalability, and fault isolation.

A hierarchical design like the one proposed here helps restrict the size of broadcast domains, improving convergence, easing deployments, and reducing the scope of failure domains. This is done by isolating a VLAN to a single wiring closet or single switch. As a result, better convergence and load-balancing upstream can be achieved through the use of L3 protocols, with no need for STP and redundancy protocols such as HSRP and VRRP. In addition, L3 designs are not subject to the same bandwidth and cable plant constraints as L2 designs; and failures are typically confined to a neighbor or route loss, instead of impacting entire broadcast domains like in L2 designs. In cases where L3 to the edge is not viable, broadcast domains should still be restricted to have no loops, with no blocked links, and each access switch having its own, unique VLANs

Figure 7-1 Hierarchical Design



Spanning Tree Protocol Security

Spanning Tree Protocol (STP) is a link management protocol, defined in the IEEE 802.1D, for bridged networks. STP provides path redundancy while preventing undesirable loops in networks consisting of multiple active paths.

Loops occur when multiple active paths exist between hosts, and which could result in an endless loop of traffic in the LAN that could bring the network down. STP implements an algorithm that guarantees a loop-free topology. With STP, all switches and bridges in the LAN exchange BPDU messages containing topology information. The STP algorithm uses the topology information to build a topological tree where only one active path exists at a time between any two hosts. Redundant paths are shutdown and used as backups in case the primary paths fail. Changes to the physical topology normally trigger a recalculation of the topological tree.

A newer version of STP exists, called Rapid-STP, is defined in IEEE 802.1w. Rapid-STP (RSTP) works similarly to STP but provides better convergence after a failure of a switch, switch port, or a LAN. RSTP significantly reduces the time to reconfigure the active topology of the network when changes occur to the physical topology or its configuration parameters. RSTP supersedes STP specified in 802.1d, but remains compatible with STP.

STP is a useful protocol but, unfortunately, both versions of the protocol were conceived with no security in mind and, as a result, are both vulnerable to several types of attacks. STP does not implement any authentication and encryption to protect the exchange of BPDUs. Because of the lack of authentication, anyone can speak to a STP-enabled device. An attacker could very easily inject bogus BPDUs, triggering a topology recalculation. A forced change to the STP topology could lead to a denial of service condition, or leave the attacker as a man-in-the-middle. In addition, because BPDUs are not encrypted, it is fairly simple to intercept BPDUs in transit, revealing important topology information.

STP introduces some security risks but, in topologies where a loop-free design is not possible, STP should be used along with the Cisco features (see Table 7-3) developed to address its risks. Not using STP would result in a loop becoming another attack vector.

Table 7-3 STP Security Features

STP Attacks and Vulnerabilities	Attack Objectives and Risk	Possible Countermeasures
Illegitimate trunk		• Disable Dynamic Trunking
STP spans VLANs	Attack on one VLAN impacts all other VLANs	• Restrict STP domain using Per-VLAN Spanning Tree (PVST)
Unauthorized spanning tree participation Bogus BPDU packets Superior BPDUs sent to become root bridge	Network instability Attacker sees frames he should not Can be used for MITM, DoS, etc	• BPDU guard • Root Guard

Cisco IOS offers a number of features that help protect bridged networks using STP against the common attacks. The following are the recommended best practices:

- Disable VLAN dynamic trunk negotiation trunking on user ports
- Use Per-VLAN Spanning Tree (PVST)
- Configure BPDU Guard
- Configure STP Root Guard
- Disable unused ports and put them into an unused VLAN
- Implement Port Security

  See Chapter 4, “Device Resiliency and Survivability.”
- Enable Traffic Storm Control

  See Chapter 4, “Device Resiliency and Survivability.”

Disable Dynamic Trunking

Dynamic trunk negotiation is a feature that facilitates the deployment of switches by an interface automatically configuring itself as a trunk according to the interface type of its neighboring. However, this feature can easily be abused to set up an illegitimate trunk. For this reason, dynamic trunking should be disabled on all ports connecting to end users.

Cisco IOS, by default, sets an interface to dynamic negotiation mode. This can be enabled using the command `switchport mode` and setting the port mode type to `access`. An example is shown in the following:

```
Router(config)# interface type slot/port
Router(config-if)# switchport mode access
```

The configuration makes the port a non-trunking, non-tagged single VLAN Layer 2 interface.

Note Catalyst 6500 switches running Cisco IOS software support the macro command `switchport host`. The `switchport host` macro command was designed to facilitate the configuration of switch ports that connect to end stations. Entering this command sets the switch port mode to access, enables spanning tree PortFast, and disables channel grouping, all at the same time. The `switchport host` macro command can be used as an alternative to the `switchport mode access` command.

For more information on the **switchport mode** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.1E/native/command/reference/S1.html#wpxref79609

For more information on the **switchport mode** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/31sg/command/reference/snmp_vtp.html#wp1210450

Per VLAN Spanning Tree (PVST)

Per-VLAN Spanning Tree (PVST) is a feature available on Catalyst 6500 and 4500 Series switches that implements a separate instance of spanning tree for each VLAN configured in the network. Having a separate instance of STP per VLAN makes the network more resilient to attacks against spanning tree. If a problem occurs in one VLAN, the effects are contained in that VLAN, shielding the rest of the network.

There are different versions of PVST, but which all maintain separate spanning tree instances per VLAN, and work in a similar fashion. Per VLAN Spanning Tree (PVST) is the original version, and which uses ISL trunking. Per VLAN Spanning Tree Plus (PVST+) provides the same functionality as PVST using 802.1Q trunking technology rather than ISL. PVST+ is an enhancement to the 802.1Q specification and is not supported on non-Cisco devices. Rapid-Per-VLAN-Spanning Tree (Rapid-PVST+) is another version of PVST that provides faster convergence of the spanning tree by using Rapid Spanning Tree Protocol (RSTP) with the existing configuration for PVST+.

PVST is enabled by default in Cisco IOS and it is recommended that PVST is always enabled.

On a Catalyst 6500 or 4500 running Cisco IOS, the default spanning tree protocol is PVST+. Rapid-PVST+ is also supported on these platforms.

In Cisco IOS, the Spanning Tree mode can be modified using the **spanning-tree mode** command. For example, to configure Rapid-PVST+:

```
Router(config)# spanning-tree mode rapid-pvst
```

For more information on the **spanning-tree mode** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.1E/native/command/reference/S1.html#wp1179548

For more information on the **spanning-tree mode** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/31sg/command/reference/snmp_vtp.html#wp1144173

BPDU Guard

Since STP does not implement any authentication or encryption to protect the exchange of BPDUs, it is vulnerable to unauthorized participation and attacks, as highlighted earlier. Cisco IOS offers the BPDU Guard feature to restrict participation in spanning tree.

End user ports should not be participating in spanning tree and, by enabling BPDU Guard on these ports, the port is shutdown if a BPDU is received. In this way, BPDU guard helps prevent unauthorized participation in spanning tree and the injection of forged BPDUs.

BPDU can be configured per port or globally. When configured globally, BPDU Guard is only effective on ports in the operational PortFast state.

BPDU Guard requires STP PortFast to be already configured on a port.

In Cisco IOS, BPDU Guard can be enabled on an interface by enabling PortFast and then using the **spanning-tree bpduguard** command as follows:

```
Router(config)# interface fastethernet 3/1
Router(config-if)# spanning-tree portfast
Router(config-if)# spanning-tree bpduguard enable
```

In Cisco IOS, BPDU Guard can be enabled globally by using the **spanning-tree portfast bpduguard default** command as follows:

```
Router(config)# spanning-tree portfast bpduguard default
```

When enabled globally, BPDU Guard applies to all interfaces that are in an operational PortFast state.

For more information on the **spanning-tree bpduguard** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.1E/native/command/reference/S1.html#wp1179312

For more information on the **spanning-tree bpduguard** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/31sg/command/reference/snmp_vtp.html#wp1065041

For more information on the **spanning-tree portfast bpduguard default** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.1E/native/command/reference/S1.html#wp1180500

For more information on the **spanning-tree portfast bpduguard default** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/31sg/command/reference/snmp_vtp.html#wp1072464

STP Root Guard

As highlighted earlier, since STP does not implement any authentication or encryption to protect the exchange of BPDUs, it is vulnerable to unauthorized participation and attacks. Cisco IOS offers the STP Root Guard feature to enforce the placement of the root bridge.

STP root guard forces a port to become a designated port so that no switch on the other end of the link can become a root switch. If a port configured for root guard receives a superior BPDU, the port it is received on is blocked. In this way, STP root guard blocks other devices from trying to become the root bridge.

STP root guard should be enabled on all ports that will never connect to a root bridge, for example, all end user ports. This ensures that a root bridge will never be negotiated on those ports.

STP root guard requires STP PortFast to be already configured on a port. STP root guard is configured on a per-port basis.

In Cisco IOS, STP Root Guard can be enabled on an interface using the **spanning-tree guard root** command as follows:

```
Router(config)# interface fastethernet 3/1
Router(config-if)# spanning-tree guard root
```

Note Do not enable loop guard and root guard on a port at the same time. Root guard forces a port to always be designated as the root port. Loop guard is effective only if the port is a root port or an alternate port.

For more information on the **spanning-tree guard root** command on the Catalyst 6500, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.1E/native/command/reference/S1.html#wp1179394

For more information on the **spanning-tree guard root** command on the Catalyst 4500, refer to the following URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/31sg/command/reference/snmp_vtp.html#wp1031770

VLAN Best Common Practices

VLAN hopping is an attack vector which provides a client with unauthorized access to other VLANs on a switch. This type of attack can be easily mitigated by applying the following best common practices:

- Always use a dedicated VLAN ID for all trunk ports
- Disable all unused ports and put them in an unused VLAN
- Do not use VLAN 1 for anything
- Configure all user-facing ports as non-trunking (DTP off)
- Explicitly configure trunking on infrastructure ports
- Use all tagged mode for the native VLAN on trunks and drop untagged frames

- Set the default port status to "disable"

CHAPTER 8

Getting Started with Security Baseline

This chapter explains how to get started with the security baseline. It presents an initial subset of tools and best practices that provide most value and with a minimal operational impact. The reader may use this chapter as a starting point, and later continue implementing the more advanced tools also part of the security baseline. This chapter provides detailed guidance on how to implement each tool and best practice, along with plenty of templates and examples.

Infrastructure Device Access

The tools and best practices here described apply to all routing and switching infrastructure devices.

Protect Local Passwords

As described in Restrict Infrastructure Device Management Accessibility, page 2-3, infrastructure devices always have local passwords and secret information that need to be properly secured. In addition to enforcing a strong password policy, secret information and password should be protected with the use of encryption.

Step 1 Global local password encryption: enable automatic password encryption with the **service password-encryption** global command. Once configured, all passwords are encrypted automatically, including passwords of locally defined users.

```
Router(config)# service password-encryption
```

Step 2 Enable secret: Define a local enable password using the **enable secret** global command. Enable access should be handled with an AAA protocol such as TACACS+ or RADIUS. The locally configured enable password will be used as a fallback mechanism after AAA is configured.

```
Router(config)# enable secret <strong-password>
```

Step 3 Line passwords: define a line password for each line you plan to use to administer the system. Note that line passwords are used for initial configuration and are not in effect once AAA is configured. Also note that some devices may have more than 5 VTYs.

```
line vty 0 4
 password <strong-password>
```

Implement Notification Banners

With the guidance of a legal professional create and apply a login banner. Login banner examples are provided in Appendix A, "Sample Configurations."

```
banner login #
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED
You must have explicit, authorized permission to access or configure this device.
Unauthorized attempts and actions to access or use this system may result in civil and/or
criminal penalties.
All activities performed on this device are logged and monitored.
#
```

AAA Services

AAA is the primary and recommended method for access control. All management access (SSH, telnet, HTTP and HTTPS) should be controlled with AAA. Point to TACACS+ and RADIUS templates. TACACS+ and RADIUS configurations templates are provided in Appendix A, "Sample Configurations."

Step 1 Enable AAA: Enable AAA with the **aaa new-model** global command. Configure **aaa session-id common** to ensure the session ID is maintained across all authentication, authorization, and accounting packets in a session.

```
aaa new-model
!
aaa session-id common
```

Step 2 Define server groups: Set server groups of all AAA servers. If possible, use a separate key per server. Set source IP address for TACACS+ or RADIUS communications.

```
tacacs-server host <TAC+server1> single-connection key <strong-key>
tacacs-server host <TAC+server2> single-connection key <strong-key>
radius-server host <RADserver1> auth-port 1645 acct-port 1646 key <strong-key>
radius-server host <RADserver2> auth-port 1645 acct-port 1646 key <strong-key>
!
aaa group server tacacs+ <TACACS-group>
 server <TAC+server1>
 server <TAC+server2>
!
aaa group server radius <RADIUS-group>
 server <RADserver1>
 server <RADserver2>
!
! Define the source interface to be used to communicate with the TACACS+/RADIUS servers
ip tacacs source-interface <Loopback or OOB interface>
ip radius source-interface <Loopback or OOB interface>
```

Step 3 Enforce login authentication: Define a login authentication method list and apply it to console, VTY and all used access lines. Use RADIUS or TACACS+ as the primary method, and local authentication as fallback. Do not forget to define a local user.

```
aaa authentication login <authen-exec-list> group <AAA-group> local-case
!
line con 0
```

```
 login authentication <authen-exec-list>
!
line vty 0 4
 login authentication <authen-exec-list>
!
```

Step 4 Enforce enable authentication: Authenticate enable access with TACACS+ or RADIUS, and use local enable as fallback method. If using TACACS+, configure a TACACS+ enable password per user. If using RADIUS, create a user named *$enab15$* and set the enable password for it. RADIUS uses this special username for enable authentication.

Note Enable access can be automatically granted as a result of exec authorization. To that end, RADIUS and TACACS+ user or group profiles need to be configured to set the privilege level to 15.

a**aa authentication enable default group** <*AAA-group*> **enable**

Step 5 Enforce exec authorization: Configure exec authorization to ensure access only to users whose profiles are configured with administrative access. TACACS+ profiles are configured with the Shell (Exec) attribute, while RADIUS users must have the service-type attribute (attribute 6) set to Administrative or Login. Define fallback method; use local if local usernames are configured with privilege level, or if-authenticated otherwise.

To grant automatic enable access to a TACACS+ user configure the user or group profile with the “privilege level” attribute to 15.

To grant automatic enable access to a RADIUS user do one of the following in the user or group profile:

- Define a Cisco av pair shell:priv-lvl=15
- Set a service-type (RADIUS attribute 6) to Administrative

```
aaa authorization exec author-exec-list group <AAA-group> if-authenticated
line vty 0 4
 authorization exec <author-exec-list>
```

Step 6 Enable accounting: Activate **exec accounting** to monitor shell connections. Enable **accounting** command for the privilege levels in used. Activate system accounting for system-level events.

```
aaa accounting send stop-record authentication failure
aaa accounting exec default start-stop group tacacs-group
aaa accounting commands 15 default start-stop group tacacs-group
aaa accounting system default start-stop group tacacs-group
```

Administrative Access

Step 1 Telnet access: If possible use SSH rather than Telnet. Telnet access can be controlled by restricting line to Telnet only, by applying ACLs controlling the sources, and by setting line timeouts. These practices are explained next. Refer to Telnet template.

Step 2 SSH access: Enable SSH access when available. SSH access can be controlled by restricting line to ssh only, by applying ACLs controlling the sources, and by setting line timeouts. These practices are explained next. Refer to SSH template.

```
! Configure a hostname and domain name
Router(config)# hostname <router-host-name>
```

```
Router (config)# ip domain-name <domain-name>

! Generate an RSA key pair, automatically enabling SSH.
Router (config)# cry key generate rsa

! Configure time-out and number of authentication retries.
Router (config)# ip ssh time-out 60
Router (config)# ip ssh authentication-retries 2
```

Step 3 HTTP access: if possible use HTTPS instead of clear HTTP. Refer to HTTP template.

```
aaa authentication login default group tacacs-group local-case
aaa authorization exec default group tacacs-group local

ip http server
ip http access-class 22
ip http authentication aaa

access-list 22 permit 172.26.0.0 0.0.255.255
access-list 22 deny   any log

line vty 0 3
 transport input telnet
```

Note HTTP access uses default login authentication and default exec authorization. In addition, privilege level for the user must be set to level 15.

Step 4 HTTPS access: Refer to HTTPS template.

```
aaa authentication login default group tacacs-group local-case
aaa authorization exec default group tacacs-group local

no ip http server
!
ip http secure-server
!
ip http access-class 22
ip http authentication aaa

access-list 22 permit 172.26.0.0 0.0.255.255
access-list 22 deny   any log

line vty 0 3
 transport input telnet
```

Restricting Access Lines and Protocols

SSH and Telnet configuration templates and examples are provided in Appendix A, “Sample Configurations.”

Step 1 Disable unnecessary access lines: disabled those ports that are not going to be used with the **no exec** line command.

```
line aux 0
 no exec
```

Step 2 Restrict incoming and outgoing protocols: per used line, explicitly define the protocols allowed for incoming and outgoing sessions. Restricting outgoing sessions prevent the system from being used as an staging host for other attacks.

```
line vty 0 4
 transport input ssh
 transport output ssh
 transport preferred none
```

Step 3 Restrict sources with ACL: Use access-class ACLs to control the sources from which sessions are going to be permitted. The source is typically the subnet where administrators reside. Use extended ACLs when available and indicate the allowed protocols. Reserve the last VTY available for last resort access. Configure an access-class to ensure this VTY is only accessed by known trusted systems.

```
! Grants access from management subnet. Set port to 22 for SSH and 23 for telnet.
access-list 111 permit tcp <management-subnet> <inverse-mask> any eq <port>
access-list 111 deny ip any any log-input

! ACL for last resort access
access-list 112 permit tcp host <management-station> any eq <port>
access-list 112 deny ip any any log-input

line vty 0 3
 access-class 111 in
!
line vty 4
 access-class 112 in
```

Step 4 Set idle and session timeouts: set idle and session timeouts in every used line. Enable tcp-keepalives to detect and close hung sessions.

```
service tcp-keepalives-in
!
line vty 0 4
 session-timeout 3
 exec-timeout 10 0
```

Routing Infrastructure

EIGRP and OSPF configuration examples and templates are provided in Appendix A, “Sample Configurations.”

Restrict Routing Protocol Membership

Step 1 Neighbor authentication: Enable neighbor authentication on all routers. Use different keys when peering with partners or other external entities. Point to EIGRP and OSPF templates.

```
! OSPF MD5 authentication
interface <interface-type/number>
  ip ospf message-digest-key <key-number> md5 <strong-password>
!
router ospf <process>
  network <network> <mask> area <area-number>
  area <area-number> authentication message-digest

! EIGRP authentication
```

```
key chain <key-chain-name>
 key 1
  key-string <strong-password>
!
interface <interface-type/number>
 ip authentication mode eigrp <process> md5
 ip authentication key-chain eigrp <process> <key-chain-name>
!
router eigrp <process>
 network <network>
!
```

Step 2 Static peer definitions: If using EIGRP, configure static peers on broadcast segments (i.e., Ethernet media) that may be subject to the intentional or unintentional insertion of bogus routers. The insertion of bogus routers is more likely on network segments with limited physical control, for example at remote locations. Balance between the perceived value and the operational burden that will result from maintain static peers.

```
router eigrp <process>
 network <network>
 neighbor <peer-address> <interface-type/number>
```

Step 3 Default passive interface: In some scenarios you may need to enable EIGRP or OSPF on a large number of interfaces where you don’t expect any routing peers to connect. Enabling the routing protocol on such interfaces may be needed to propagate directly connected networks. In these scenarios you may want to enable the routing protocol on a network range matching multiple interfaces. If you don’t expect routing peers to connect to the majority of these networks, you may want to disable the propagation of routing updates by default by using the passive-interface default command. Once this command is configured, you need to explicitly enable routing updates on the interfaces where you expect routing peers.

```
router <protocol> <process>
! Disable routing updates on all interfaces by default
 passive-interface default
! Explicitly enable routing updates on interfaces where you expect routing peers
 no passive-interface <interface-type/number>
```

Step 4 BGP TTL Security Check: If using BGP, configure the TTL security check on routers connecting to external BGP peers.

```
router bgp <as-number>
 neighbor <ip-address> ttl-security hops <hop-count>
```

Route Filtering

To implement route filtering, follow these steps:

Step 1 Implement peer prefix filtering at the edges: Implement inbound filters at the edges to ensure only the expected routes are introduced into the network. Balance between higher control and associated operational burden. Deploy filters at edges where invalid routing information may be most likely introduced from, example the at the WAN edge. Controlling incoming routing updates at the WAN edge not only mitigates the introduction of bogus routes at the branches, but it also prevents a dual access branch from becoming a transit network.

```
! Incoming route filter applied at the WAN edge and that only allows the branch subnet.
!
```

```
router eigrp <process>
 network <network>
 distribute-list 39 in <interface-type/number>
!
access-list 39 permit <remote-subnet> <inverse-mask>
```

Step 2 Enforce route filters at stub routers: At branches and remote locations with stub networks, enforce route filters to prevent the propagation of invalid routing information.

If using EIGRP, use the eigrp stub connected command to ensure propagation of directly connected networks only.

```
router eigrp <process>
 network <network>
 eigrp stub connected
```

If using other protocols, use outbound filters:

```
! Outbound route filter applied at the branch router.
!
router ospf <process>
 distribute-list 33 out <interface-type/number>
!
access-list 33 permit <branch-subnet> <inverse-mask>
```

Step 3 Neighbor logging: On all routers enable the logging of status changes of neighbor sessions.

```
!Logging neighbor changes in EIGRP
router eigrp <process>
 eigrp log-neighbor-changes

! Logging neighbor changes in OSPF
router ospf <process>
 log-adjancey-changes
```

Device Resiliency and Survivability

Disabling Unnecessary Services

To disable services that are not needed, follow these steps:

Step 1 Identify Open Ports: Use the show control-plane host open-ports command to see what UDP/TCP ports the router is listening to, and determine which services need to be disabled.

```
cr18-7200-3#show control-plane host open-ports
Active internet connections (servers and established)
Prot        Local Address      Foreign Address                  Service    State
 tcp                 *:22                  *:0               SSH-Server   LISTEN
 tcp                 *:23                  *:0                   Telnet   LISTEN
 tcp              *:63771    172.26.150.206:49         IOS host service ESTABLIS
 udp                 *:49     172.26.150.206:0           TACACS service   LISTEN
 udp                 *:67                  *:0            DHCPD Receive   LISTEN

cr18-7200-3#
```

Note The show **control-plane host open-ports** command was introduced in Cisco IOS Release 12.3(4)T. For earlier versions, use the **show ip sockets** command to identify open UDP ports, and the **show tcp brief all** and **show tcp tcb** commands to see open TCP ports. For more information, check the Open Ports and Sockets section under Network Telemetry.

Step 2 Global services disabled by default: Unless explicitly needed, ensure finger, identification (identd), and TCP and UPD small servers remain disabled on all routers.

```
! Global Services disabled by default
Router(config)# no ip finger
Router(config)# no ip identd
Router(config)# no service tcp-small-servers
Router(config)# no service udp-small-servers
```

Note that all these services are disabled by default, their configurations do not appear in the CLI unless enabled.

Step 3 Global services enabled by default: Unless explicitly needed, BOOTP, IP Source Routing, and PAD services should be disabled globally on all routers.

```
! Disable BOOTP, IP Source Routing and PAD global services
Router(config)# no ip source-route
Router(config)# no ip bootp server
Router(config)# no service pad
```

Note that, because DHCP is based on BOOTP, both of these services share UDP server port 67 (per RFC 951, RFC 1534, and RFC 2131).

Also, note that these services are enabled by default and their configurations do not appear in the CLI unless disabled.

Step 4 IP directed broadcast: Make sure directed broadcasts remain disabled on all interfaces.

```
! Disable IP  directed broadcasts on all interfaces
Router(config)# interface <interface-type/number>
Router(config-if)# no ip directed-broadcast
```

Note that IP directed broadcasts is disabled by default, its configuration does not appear in the CLI unless enabled.

Step 5 When to disable CDP: Disable CDP on interfaces where the service may represent a risk, for example on external interfaces such those at the Internet edge.

```
! Disable CDP on externally facing interfaces
Router(config)# interface <interface-type number>
Router(config-if)# no cdp enable
```

Note that CDP is enabled by default and its configuration does not appear in the CLI unless disabled. To ensure CDP is disabled on an interface, either use the **show cdp interface** command or check if the interface configuration contains the **no cdp enable** command.

In the following example CDP has been left enable on interface FastEthernet 2/1, and it was explicitly disabled on FastEthernet 2/0:

```
Router#show cdp interface FastEthernet 2/1
FastEthernet2/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Routershow cdp interface FastEthernet 2/0
```

```
Router#
Router #sh run int fastEthernet 2/0
Building configuration...

Current configuration : 163 bytes
!
interface FastEthernet2/0
ip address 198.133.219.5 255.255.255.0
no cdp enable
end
```

Step 6 Access and externally facing ports: Unless required, disable MOP, IP Redirects, and Proxy ARP on all access and externally facing interfaces. This typically includes access lines at campuses and branches, and externally-facing ports such those at the Internet edge.

```
! Disable MOP, IP Redirects,
Router(config)# interface <interface-type/number>
Router(config-if)# no mop enabled
Router(config-if)# no ip redirects
Router(config-if)# no ip proxy-arp
```

Dual-homed campuses and branches commonly implement HSRP or other First Hop Routing Protocol; in such scenarios IP redirect is typically not needed.

Infrastructure Protection ACLs (iACLs)

As discussed earlier in this document, iACLs are useful when deployed at the network edges (i.e., peering points for ISPs, and network boundaries within enterprise networks). From all the inner network edges existent in an enterprise network, the Internet edge and the WAN edge are probably the best places to start configuring an iACL. This section discusses the process for implementing iACLs using the WAN edge as an example, but the same principles can be used for any other placement. For more iACL examples, refer to Appendix A, "Sample Configurations."

iACLs require a clear understanding of the protocols and ports legitimately used by the infrastructure; therefore, it is highly recommended that you start by using a discovery ACL. The discovery ACL is an advanced ACL with entries for each of the expected protocol and ports. At the end, the discovery ACL must have an explicit entry permitting any other IP traffic. All entries must be ideally configured with the log option to generate logs for each match. Since the purpose of this ACL is to identify the traffic patterns and not to control access, all entries are configured to permit all matching traffic for any sources and destinations. Once designed, the discovery ACL must be activated in the same interfaces where the iACL will be configured.

The following is an example of the discovery ACL used in our lab network and designed with the expected protocols and ports. Overall, the more granular the better, so that you should define individual entries for each traffic you expect in your network. Note that the logging of ACL entries may affect performance; therefore, you may want to enable log for one entry at the time to reduce the impact. Also note that discovery ACL ends with an explicit permit for all other traffic. This entry is configured to catch up any traffic not yet identified.

```
access-list 133 permit eigrp any any log
access-list 133 permit icmp any any log
access-list 133 permit tcp any any eq tacacs log
access-list 133 permit udp any any eq ntp log
access-list 133 permit tcp any any eq 22 log
access-list 133 permit udp any any eq syslog log
access-list 133 permit ip any any log
```

Once the discovery ACL is designed, it should be activated in the same interface where the iACL will reside. Make sure the ACL is configured as inbound. In our example, we define the discovery ACL on the WAN edge router as follows:

```
interface GigabitEthernet4/46
 description To Branch 4
 ip address 10.139.5.10 255.255.255.254
 ip access-group 133 in
```

Periodically monitor the activity on the discovery ACL by using the **show access-list** and the **show logging** commands. The **show access-list** command will show if any packets have been seen for each one of the identified protocols and ports.

```
cr18-7604-1#sh access-l 133
Extended IP access list 133
    10 permit eigrp any any log (76 matches)
    20 permit icmp any any log
    30 permit tcp any any eq tacacs log (10 matches)
    40 permit udp any any eq ntp log (8 matches)
    50 permit tcp any any eq 22 log (35 matches)
    60 permit udp any any eq syslog log (4 matches)
    70 permit ip any any log (24 matches)
```

Matches in the final permit **any any** entry may indicate that there are still protocols and ports directed to the infrastructure that are yet not identified. At this point, you may want to check the logging records to see what packets are matching the last ACL entry. As regular user traffic will also match the last entry, you should only pay attention to packets destined to the infrastructure address space.

```
cr18-7604-1#sh logging
...
%SEC-6-IPACCESSLOGP: list 133 permitted tcp 10.139.5.11(55786) -> 172.26.150.206(49), 1 
packet
%SEC-6-IPACCESSLOGDP: list 133 permitted icmp 10.139.5.11 -> 172.26.159.166 (8/0), 1 
packet
%SEC-6-IPACCESSLOGP: list 133 permitted tcp 10.139.5.11(22) -> 172.26.159.166(11031), 1 
packet
%SEC-6-IPACCESSLOGRP: list 133 permitted eigrp 10.139.5.11 -> 224.0.0.10, 34 packets
%SEC-6-IPACCESSLOGP: list 101 permitted udp 10.139.5.11(123) -> 172.26.158.236(123), 1 
packet
%SEC-6-IPACCESSLOGDP: list 133 permitted icmp 10.139.5.11 -> 172.26.159.166 (8/0), 4 
packets
%SEC-6-IPACCESSLOGP: list 133 permitted tcp 10.139.5.11(22604) -> 172.26.150.206(49), 19 
packets
%SEC-6-IPACCESSLOGP: list 133 permitted tcp 10.139.5.11(55786) -> 172.26.150.206(49), 5 
packets
%SEC-6-IPACCESSLOGP: list 133 permitted tcp 10.139.5.11(22) -> 172.26.159.166(11031), 31 
packets
%SEC-6-IPACCESSLOGRP: list 133 permitted eigrp 10.139.5.11 -> 224.0.0.10, 8 packets
%SEC-6-IPACCESSLOGP: list 133 permitted tcp 10.139.5.11(13621) -> 172.26.150.206(49), 2 
packets
%SEC-6-IPACCESSLOGRP: list 133 permitted eigrp 10.139.5.11 -> 224.0.0.10, 1 packet
%SEC-6-IPACCESSLOGP: list 133 permitted tcp 10.139.5.11(22) -> 172.26.159.166(11032), 1 
packet
%SEC-6-IPACCESSLOGP: list 133 permitted tcp 10.139.5.11(40429) -> 172.26.159.167(22), 1 
packet
%SEC-6-IPACCESSLOGP: list 133 permitted tcp 10.139.5.11(22604) -> 172.26.150.206(49), 7 
packets
%SEC-6-IPACCESSLOGP: list 101 permitted udp 10.139.5.11(58557) -> 172.26.150.206(514), 4 
packets
```

The **show logging** command also shows the exact sources, destinations, and ports for the traffic flows. This provides the more granular information needed to craft the iACL. You may want to start configuring more specific entries in your discovery ACL to validate the information learned:

```
access-list 133 permit eigrp host 10.139.5.11 host 224.0.0.10
access-list 133 permit icmp host 10.139.5.11 172.26.0.0 0.0.255.255
access-list 133 permit tcp host 10.139.5.11 eq 22 172.26.0.0 0.0.255.255
access-list 133 permit tcp host 10.139.5.11 172.26.0.0 0.0.255.255 eq 22
access-list 133 permit tcp host 10.139.5.11 host 172.26.150.206 eq 49
access-list 133 permit udp host 10.139.5.11 eq ntp host 172.26.158.236 eq ntp
access-list 133 permit udp host 10.139.5.11 host 172.26.150.206 eq syslog
access-list 133 permit ip any any log
```

With all the information at hand, add the necessary new entries matching the protocols and ports identified, and repeat the process until you feel confident that you have identified all traffic destined to the infrastructure address space.

Step 1 Now that you should have gained a good understanding on the control and management plane traffic you may start crafting the first module of the iACL. Start by defining antispoofing entries protecting the infrastructure address space. In our example, network 172.26.0.0/16 is reserved to the OOB network, and therefore no packets in this range should be originated from any of the branches.

```
!--- Module 1: Anti-spoofing, deny special use addresses
! Deny your OOB address space as a source in packets
access-list 101 deny ip 172.26.0.0 0.0.255.255 any
```

Also define the necessary entries to block packets with IP source addresses that would be invalid for the given scenario. In our case branches are configured with private addresses; therefore, those IP address are expected and for this reason we do not include deny entries blocking those ranges.

```
!--- Deny special-use address sources.
!--- See RFC 3330 for additional special-use addresses.
access-list 101 deny ip host 0.0.0.0 any
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
access-list 101 deny ip 192.0.2.0 0.0.0.255 any
access-list 101 deny ip 224.0.0.0 31.255.255.255 any
```

Step 2 In the second module include the necessary entries to explicitly permit the traffic learned with the discovery ACL. Be as specific as possible.

```
!--- Module 2:  Explicit Permit
! Permit valid traffic destined to the infrastructure
access-list 101 permit eigrp host 10.139.5.11 host 224.0.0.10
access-list 101 permit icmp host 10.139.5.11 172.26.0.0 0.0.255.255
access-list 101 permit tcp host 10.139.5.11 eq 22 172.26.0.0 0.0.255.255
access-list 101 permit tcp host 10.139.5.11 172.26.0.0 0.0.255.255 eq 22
access-list 101 permit tcp host 10.139.5.11 host 172.26.150.206 eq 49
access-list 101 permit udp host 10.139.5.11 eq ntp host 172.26.158.236 eq ntp
access-list 101 permit udp host 10.139.5.11 host 172.26.150.206 eq syslog
```

Step 3 Next, deny any other access to the infrastructure. In our example, we block any other traffic destined to three networks used in the infrastructure: the 10.139.5.0/24 network is used for the WAN links between the edge and the branches; 10.122.0.0/16 is used on all core equipment; and 172.26.0.0/16 is reserved to the OBB network. Valid traffic to these networks should be allowed by the second module.

```
!--- Module 3:  Explicit Deny to Protect Infrastructure
! Deny all other access to infrastructure
access-list 101 deny ip any 10.139.5.0 0.0.0.255
```

```
access-list 101 deny ip any 10.122.0.0 0.0.255.255
access-list 101 deny ip any 172.26.0.0 0.0.255.255
```

Step 4 Finally, and depending on your requirements, configure a final ACL entry to either permit or deny any other traffic. In our example we are building an WAN edge iACL, which objective is to control traffic destined to the infrastructure, while allowing any other traffic coming from the branches. Therefore we use a permit any any entry. Please remember that other scenarios may require the configuration of a deny any any.

```
!--- Module 4:  Explicit Permit/Deny for Transit Traffic
! Permit transit traffic enterprise inner iACL
access-list 101 permit ip any any
```

For complete configuration and additional iACL examples, refer to Appendix A, “Sample Configurations.”

Port Security

Port security is an useful feature for mitigating MAC flooding and other Layer 2 Content Addressable Memory (CAM) overflow attacks. The access edges and the server farms are the first places in the network where port security can provide most value. The access edges at campuses and remote offices provide the first line of connectivity to users, and therefore are prone to MAC flooding and other Layer 2 attacks. Server farms, including Demilitarized Networks (DMZs), are accessed by a wide range of users, and therefore are also subject to the same attacks.

To implement port security, follow these steps:

Step 1 Port Security at the access edge: Port Security may be configured at the access switches at campuses and branch offices to limit the maximum number of MAC addresses allowed per port. Since the number of MAC addresses to be allowed may vary depending on the system or application attached to the switch port, it is highly important you do your homework first. A good way to identify the MAC addresses per port is by checking the CAM tables with the **show mac-address-table** command, as shown in the following example:

```
r17-3750-2#show mac-address-table  vlan 20
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 All    0100.0ccc.cccc    STATIC      CPU
...
 All    ffff.ffff.ffff    STATIC      CPU
  20    000f.352c.4bd1    DYNAMIC     Gi2/0/48
  20    0014.a92e.7737    DYNAMIC     Gi2/0/47
  20    0015.627f.abb0    DYNAMIC     Gi2/0/47
  20    0015.629c.2f33    DYNAMIC     Gi2/0/47
Total Mac Addresses for this criterion: 25
cr17-3750-2#
```

As a rule of thumb, switch ports connecting to individual workstations must be restricted to a single MAC address, while ports connecting to IP phones should be limited to two MAC addresses.

In addition, as in access edges it is difficult to tell what MAC address will connect to which port, port security is best configured with dynamic MAC learning. It is also a good practice to start with a conservative response action, either protect or restrict, and not shutdown.

Switch ports used by workstations should be restricted to one host as follows:

```
Router(config)# interface gigabitethernet0/2
Router(config-if)# switchport port-security maximum 1
Router(config-if)# switchport port-security violation restrict
Router(config-if)# switchport port-security
```

IP phone ports should be restricted to two MAC addresses:

```
Router(config)# interface gigabitethernet0/1
Router(config-if)# switchport port-security maximum 2
Router(config-if)# switchport port-security violation restrict
Router(config-if)# switchport port-security
```

Step 2 Port Security at server farms and DMZs: In environments where systems hardly change location and remain connected to the same switch ports, port security can be configured to permit specific MAC addresses. This is generally the case in server farms and DMZs, where one typically knows what MAC address connects to which switch port.

The **show mac-address-table** command can be used to identify the MAC address of the system connected to the port to be configured:

```
r17-3750-2#show mac-address-table  vlan 20
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 All    0100.0ccc.cccc    STATIC      CPU
...
 All    ffff.ffff.ffff    STATIC      CPU
  20    000f.352c.4bd1    DYNAMIC     Gi2/0/48
...
Total Mac Addresses for this criterion: 25
```

Because of the fact that in static environments it is easier to track MAC addresses, one can opt for a more restrictive response action (i.e., shutdown in response to a violation).

The following example illustrates how a port can be restricted for use by only one specific host, with the defined MAC address, such as may be employed in a DMZ.

```
Router(config)# interface gigabitethernet2/0/48
Router(config-if)# switchport port-security maximum 1
Router(config-if)# switchport port-security mac-address 000f.352c.4bd1
Router(config-if)# switchport port-security violation shutdown
Router(config-if)# switchport port-security
```

Step 3 If using SNMP, enable SNMP logging of Port Security policy violations as in the example below. It is also a good practice to rate limit the generation of SNMP trap messages to ensure the availability of the system, even under attack.

```
snmp-server enable traps port-security
snmp-server enable traps port-security trap-rate <max number of traps per second>
```

Network Telemetry

Time Synchronization (NTP)

Time synchronizations is critical for event analysis and correlation, thus enabling NTP on all infrastructure components is a fundamental requirement.

When implementing NTP considered these best common practices:

- Prefer a hierarchical NTP design versus a flat design.
- Use a common, single time zone across the entire infrastructure to facilitate the analysis and correlation of events.
- Control which clients and peers can talk to an NTP server, and enable NTP authentication.

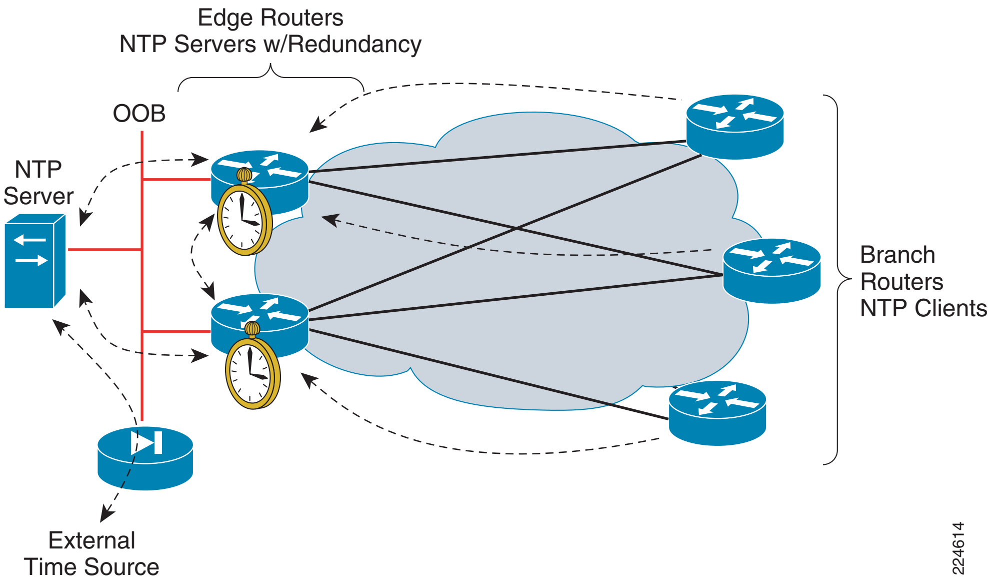
When implementing NTP follow these principles:

NTP Design: Hierarchical designs are preferred because they are highly stable, scalable and provide most consistency. A good way to design a hierarchical NTP network is by following the same structure as the routing architecture in place.

NTP Design for Remote Offices

Branch offices are typically aggregated at one or more WAN edge routers that can be leveraged in the NTP design. At the headquarters, you will likely have internal time servers at a secured segment. Unless you have an in-house atomic or GPS based clock, these internal time servers will be synchronized with external time sources. Following the routing design, the WAN edge routers may be configured as time servers with a client/server relationship with the internal time servers, and the branch routers may be configured as clients (non-time servers) with a client/server relationship with the WAN edge routers. This design is depicted in Figure 8-1.

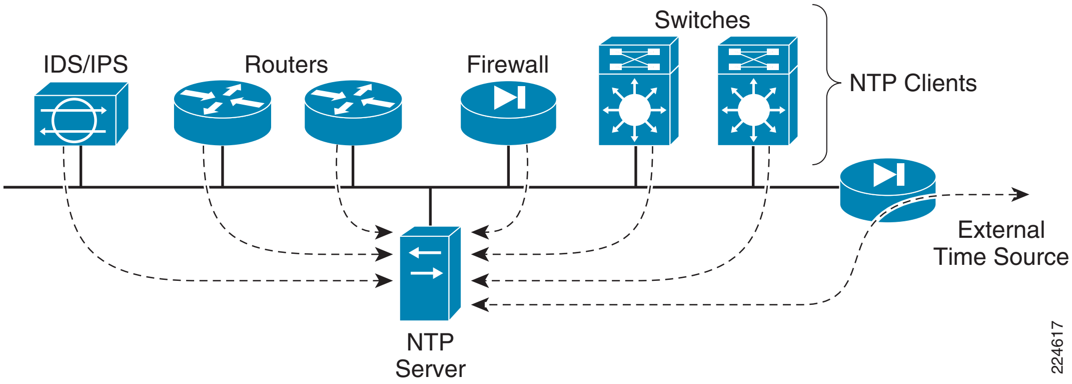
Figure 8-1 NTP Design for the WAN Edge and Remote Offices



NTP Design at the Headquarters

At the headquarters or main office, you may take advantage of an existing OOB management network. Transporting NTP over the OOB network flattens and simplifies the design. In this scenario, all routers and switches may be configured as clients (non time servers) with a client/server relationship with the internal time servers located at a secured segment. These internal time servers are synchronized with external time sources. This design is illustrated in Figure 8-2.

Figure 8-2 NTP Design Leveraging an OOB Management Network



In case there is no OOB management network, then you can copy the same structure as the routing design. The core routers may be configured as time servers with a client/server relationship with the internal servers located at a secured segment and synchronized with external sources, distribution routers may be configured as time servers with a client/server relationship with the core routers, and all other internal clients (non time servers) may be set with a client/server relationship with the distribution routers.

Step 1 NTP Servers: When configuring routers and switches as time severs follow these practices:

a. Enable timestamp information for debug and log messages:

```
Router(config)# service timestamps debug datetime localtime show-timezone msec
Router(config)# service timestamps log datetime localtime show-timezone msec
```

b. Set time-zone and summertime adjustments

```
Router(config)# clock timezone zone hours-offset [minutes-offset]
Router(config)# clock summer-time zone recurring [week day month hh:mm week day month
hh:mm [offset]]
```

c. Set the source IP address to be used for NTP packets. Use the IP address of an administrative loopback or the OOB interface.

```
Router(config)# ntp source <interface-type/number>
```

d. Restrict the IP addresses of the servers and peers this server will communicate with.

```
Router(config)# access-list 10 remark ACL for NTP Servers and Peers
Router(config)# access-list 10 permit <NTPserver1>
Router(config)# access-list 10 permit <NTPserver2>
Router(config)# access-list 10 permit <NTPpeer1>
Router(config)# access-list 10 permit <NTPpeer2>
Router(config)# access-list 10 deny any log
!
```

```
Router(config)# ntp access-group peer 10
```

e. Restrict the IP addresses of the clients that can communicate with this server.

```
Router(config)# access-list 15 remark ACL for NTP Client
    Router(config)# access-list 15 permit <Client1>
    Router(config)# access-list 15 permit <Client2>
    Router(config)# access-list 15 deny any log
    !
    Router(config)# ntp access-group serve-only 15
```

f. Enable NTP authentication

```
Router(config)# ntp authentication-key <key#> md5 <strong8charkey>
Router(config)# ntp trusted-key <key#>
Router(config)# ntp authenticate
```

g. Define the NTP servers

```
Router(config)# ntp server <NTPserver1>
Router(config)# ntp server <NTPserver2>
```

h. Define any NTP peers. For redundancy purposes it is highly recommended to deploy NTP servers in pairs.

```
Router(config)# ntp peer <NTPpeer1>
```

Refer to Appendix A, "Sample Configurations," for sample NTP configurations.

Step 2 NTP Clients: When configuring routers and switches as time clients follow these practices:

a. Enable timestamp information for debug and log messages:

```
Router(config)# service timestamps debug datetime localtime show-timezone msec
Router(config)# service timestamps log datetime localtime show-timezone msec
```

b. Set time-zone and summertime adjustments:

```
Router(config)# clock timezone zone hours-offset [minutes-offset]
Router(config)# clock summer-time zone recurring [week day month hh:mm week day month
hh:mm [offset]]
```

c. Set the source IP address to be used for NTP packets. Use the IP address of an administrative loopback or the OOB interface.

```
Router(config)# ntp source <interface-type/number>
```

d. Enable NTP authentication

```
Router(config)# ntp authentication-key <key#> md5 <strong8charkey>
Router(config)# ntp trusted-key <key#>
Router(config)# ntp authenticate
```

e. Define the NTP servers

```
Router(config)# ntp server <NTPserver1>
Router(config)# ntp server <NTPserver2>
```

Refer to Appendix A, "Sample Configurations," for sample NTP configurations.

Local Device Traffic Statistics

Routers and switches maintain per-interface and global statistics that are essential for network telemetry. This information includes per-interface throughput and bandwidth statistics, enabled features and global per-protocol traffic statistics.

By default, Cisco IOS routers calculate interface statistics based on a default 5-minute average. While the default configuration works well for most router and switch placements, a 5-minute average may not reflect sudden bursts of traffic as quickly as needed. This is particularly important for equipment deployed at edges with higher exposure, such as the Internet edge and WAN edge. To make computations be more reactive to sudden bursts of traffic, set the length of time for which data is used to compute load statistics to one minute on the edge interfaces:

```
Router(config)# interface <interface-type number>
Router(config)# load-interval 60
```

System Status Information

Routers and switches maintain an array of system resource information that reflects the overall health and status of the system. The following are best practices for all switches and routers:

Step 1 Memory Threshold: Enable memory threshold syslog notification to alert when available free memory falls below recommended levels.

A good practice is to set the free memory threshold to a 10% of the total memory. Use the **show memory** command to see the total memory and available free memory.

In this example we set a 10% threshold for both IO and processor memories:

```
Router#show memory
                Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor    6572AD00   915231348    27009876   888221472   374721396   361583220
      I/O     C000000    67108864     5856500    61252364    61233808    61232028
…
Router#
```

The total system processor memory is 915,231,348 bytes, so the threshold is set to 91,523 Kilobytes:

```
Router(config)#memory free low-watermark processor 91523
```

The total system IO memory is 67,108,864 bytes, therefore the threshold is set to 6,710 Kilobytes:

```
Router(config)#memory free low-watermark io 6710
```

Step 2 Enable critical system logging protection: When a router is overloaded by processes, the amount of available memory might fall to levels insufficient for it to issue critical notifications. Reserve a region of 1000 Kilobytes of memory to be used by the router for the issuing of critical notifications as follows:

```
Router(config)# memory reserve critical 1000
```

Step 3 Enable CPU threshold SNMP trap notification: Increases in CPU load on routers and switches often indicate an event is taking place, therefore enabling the notification of high CPU conditions is always recommended. However, keep in mind that high CPU is not always an indicator of malicious activity, and other sources of information should be considered.

To configure CPU thresholding follow these steps:

a. Enable CPU thresholding violation notification as traps and inform requests.

```
Router(config)# snmp-server enable traps cpu threshold
```

b. Sends CPU traps to the specified address.

```
router(config)# snmp-server host <snmp-server> traps public cpu
```

c. Set the CPU utilization threshold to 80 percent for a rising threshold notification and 20 percent for a falling threshold notification, with a 5-second polling interval.

```
Router(config)# process cpu threshold type total rising 80 interval 5 falling 20
interval 5
```

d. Limit the number of entries and the size of the history table for CPU utilization statistics. Generate entries only when a process exceeds 40% of CPU utilization, and retain entries for 300 seconds:

```
Router(config)# process cpu statistics limit entry-percentage 40 size 300
```

CDP Best Common Practices

CDP is enabled by default on Cisco routers and switches. The best common practices for CDP are currently:

- Enable CDP on point-to-point infrastructure inks (leave default configuration)
- Disable CDP on edge devices or interfaces where is it not required and where such as service may represent a risk, including:
  - LAN access edge
  - Data Center access edge
  - Internet transit edge
  - Extranet edge
  - Any public-facing interfaces

When and how to disable CDP is described in detail in Step 5 under Disabling Unnecessary Services, page 8-7.

System Logging (Syslog)

Syslog provides invaluable operational information, including system status, traffic statistics and device access information. For this reason, syslog is recommended on all network devices.

Follow these practices when enabling syslog:

Step 1 Enable timestamps for debugging and logging messages. Adding timestamps to messages facilitates analysis and correlation.

```
Router(config)#service timestamps debug datetime msec localtime show-timezone
Router(config)#service timestamps log datetime msec localtime show-timezone
```

Step 2 Enable system message logging to a local buffer. This allows to access the logging information directly from the router or switch in case of communication failure with the syslog server. It is important to note that local buffers are circular in nature, so newer messages overwrite older messages after the buffer is filled.

```
Router(config)# logging buffered
```

Step 3 Set the severity level of messages to be logged. Messages at or numerically lower than the specified level are logged.

```
Router(config)# logging trap <level>
```

With respect to the severity level, the more information is logged the better, so logging messages of all severity levels would be ideal. However, this may result in an overwhelming volume of messages. As a result, a balance needs to be found between desired information detail and a digestible volume of messages. If available, syslog rate-limiting helps keeping the message volume under control. Another good practice is to enable more detailed logging on critical systems or systems that may more accessible to external or remote users, such as equipment on the Internet and WAN edges, and only log critical alerts for the rest of the infrastructure.

Step 4 Logging for critical equipment: Enable logging of all severity messages, but enable the rate-limiting for messages of level 3 and up. Critical, alert, and emergency messages are not limited.

```
Router(config)# logging trap debugging
Router(config)#  logging rate-limit 1 except 3
```

Logging for non-critical equipment: Enable logging for critical, alert and emergency messages only.

```
Router(config)# logging trap critical
```

Step 5 Define login facility: Different facility numbers can be used to organized messages in different repositories. By default, Cisco routers and switches export syslog as facility local7.

```
Router(config)# logging facility <facility0-7>
```

Step 6 Define the syslog servers to be used.

```
Router(config)# logging facility <syslogserver>
```

Step 7 Set the source IP address of syslog messages to the address of an administrative loopback interface or OOB interface.

```
Router(config)# logging source-interface <interface-type number>
```

Step 8 Disable the logging of messages to the console. This helps keep the console free of messages.

```
Router(config)# no logging console
```

SNMP

SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network. It provides valuable system and event information, therefore should be enabled throughout the network infrastructure.

In case SNMP access is not required, make sure it is disabled. The **no snmp-server** command disables all running versions of SNMP (SNMPv1, SNMPv2C, and SNMPv3) on the device.

```
Router(config)# no snmp-server
```

Use the **show snmp** command to see if any SNMP agent is active:

```
Router# show snmp
%SNMP agent not enabled
Router#
```

When SNMP is required, follow these best practices:

Step 1 Restrict what systems can access the SNMP agent running on the router or switch. Be as specific as possible, for instance, only permitting access from the SNMP management stations.

```
Router(config)# access-list <ACL#> remark ACL for SNMP access to device
Router(config)# access-list <ACL#> permit <SNMPserver1>
Router(config)# access-list <ACL#> permit <SNMPserver2>
Router(config)# access-list <ACL#> deny any log
```

Go to Step 2 if planning to enforce this ACL in conjunction with SNMP views. To activate this ACL without any SNMP vies, use the following command:

```
Router(config)# snmp-server community <community-string> RO < ACL#>
```

Step 2 If using SNMPv3 (recommended), enforce an SNMP view that restricts the download of full IP routing and ARP tables

```
! Define an SNMP view which denies queries to download the full IP routing and ARP tables
Router(config)# snmp-server view <restricted-view1> internet included
Router(config)# snmp-server view <restricted-view1> ipRouteTable excluded
Router(config)# snmp-server view <restricted-view1> ipNetToMediaTable excluded
Router(config)# snmp-server view <restricted-view1> at excluded
```

In case any of the above keywords are not recognized by a particular IOS release, use the ones below

```
Router(config)# snmp-server view <restricted-view1> internet included
Router(config)# snmp-server view <restricted-view1> ip.21 excluded
Router(config)# snmp-server view <restricted-view1> ip.22 excluded
Router(config)# snmp-server view <restricted-view1> mib-2.3 excluded
```

Once the views are defined, associate them to the SNMP community strings. In addition, restrict access to those communities to read-only (RO) and apply the previously defined ACL.

Define the SNMP community strings, restricting them to read-only access, enforce a restricted view and apply the xACL

```
Router(config)# snmp-server community <community-string> view <restricted-view1> RO <ACL#>
```

Note By default, an SNMP community string permits read-only access to all objects.

Multiple community strings may be defined, each with different views and different ACLs, thereby enabling different SNMP managers to be restricted to only authorized queries

If the snmp-server community command is not explicitly defined, it is automatically added to the configuration upon entering the snmp host command. In this case, the default community string for the snmp-server community will be taken from the snmp host command.

Step 1 Define the SNMP managers, their supported SNMP version and permitted community string.

Step 2 If SNMP v3 is supported, enable only SNMP v3 and with the maximum security level supported by the SNMP managers, using encrypted communication ('priv') where feasible. The engine ID of an SNMP v3 SNMP manager is required in order to enable SNMP v3.

```
Router(config)# snmp-server engineID remote <SNMPmgrIP1> <engineID>
Router(config)# snmp-server host <SNMPmgrIP1> version 3 priv <community-string>
```

Note If no version is specified, the default is SNMP v1. If SNMP v3 is specified but no security level is defined, the default is 'noauth', with only username authentication and no encryption. SNMP v3 security levels 'noauth' and 'auth' do not encrypt SNMP communication.

If SNMP v3 is supported, define an SNMP v3 user group with security level 'authPriv' and enforce a restricted view.

```
Router(config)# snmp-server group <v3groupname> v3 priv read <restricted-view1>
```

Define a user within the SNMP v3 group, along with their authentication password and encryption key

```
Router(config)# snmp-server user <username> <v3groupname> v3 encrypted auth md5
<auth-passwd> priv des56 <enc-key>
```

Step 3 Set the source IP address for SNMP traps to the address used on the administrative loopback interface of OOB interface.

```
Router(config)# snmp-server trap-source <Loopback/OOB Interface>
```

a. Configure the system to send a trap on SNMP authentication failure. Ensure the SNMP management device is configured to react to the authentication failure trap.

```
Router(config)# snmp-server enable traps snmp authentication
```

b. Configure the System to send a trap for configuration changes.

```
Router(config)# snmp-server enable traps config
```

c. Configure the system to send a trap for environmental monitor threshold exceptions

```
Router(config)# snmp-server enable traps envmon
```

d. Enable any additional required traps. It is recommended that only operationally important traps are enabled (for example, BGP state changes). Ensure enabled traps are monitored.

```
Router(config)#  snmp-server enable traps bgp
```

Step 4 If configuration files are downloaded via SNMP by TFTP servers, restrict which TFTP servers may do so.

```
access-list <ACL#> remark ACL for TFTP
access-list <ACL#> permit host <TFTPserver1>
access-list <ACL#> permit host <TFTPserver2>
access-list <ACL#> deny any log
!
snmp-server tftp-server-list <ACL#>
```

Network Policy Enforcement

Access Edge Filtering

The implementation of access edge filtering to protect the network infrastructure itself has been already discussed in the Infrastructure Protection ACLs (iACLs), page 8-9.

uRPF

The access edge filtering techniques discussed in the Infrastructure Protection ACLs (iACLs), page 8-9, provide the basis for spoofing protection based on ACLs. Here, we explain the use of uRPF as a complementary tool. uRPF offers a dynamic technique for enabling BCP38/RFC 2827 ingress traffic filtering, with minimum operational overhead and performance impact.

The Internet edge and the access edges are two good places to start enabling uRPF.

Internet Edge

When enabling uRPF at the Internet edge follow these steps:

Step 1 Configure uRPF strict mode on the internal interfaces:

```
Router(config)# interface <Type Number>
Router(config-if)# ip verify unicast source reachable-via rx
```

Step 2 Configure uRPF loose mode on Internet facing interfaces:

```
Router(config)# interface <Type Number>
Router(config-if)# ip verify unicast source reachable-via any
```

Access Edges

Enable uRPF strict mode at the first routed hop of the access edges of campuses or remote offices:

```
Router(config)# interface <Type Number>
Router(config-if)# ip verify unicast source reachable-via rx
```

Switching Infrastructure

At your Layer 2 infrastructure follow these best practices:

Step 1 Disable dynamic trunking on all switching access lines:

```
Router(config)# interface type slot/port
Router(config-if)# switchport mode access
```

To disable dynamic trunking on a range of ports:

```
Router(config)# interface range type slot/first-port - last-port
Router(config-if)# switchport mode access
```

Step 2 Enable BPDU guard on end user ports and other ports not expected to participate in Spanning Tree:

```
Router(config)# interface type slot/port
Router(config-if)# spanning-tree portfast
Router(config-if)# spanning-tree bpduguard enable
```

To enable BPDU guard on a range of ports:

```
Router(config)# interface range type slot/first-port - last-port
Router(config-if)# spanning-tree portfast
Router(config-if)# spanning-tree bpduguard enable
```

Step 3 In some switching platforms interfaces are enabled by default. It is a good practice to disable all unused ports and place them into an unused VLAN:

```
Router(config)# interface type slot/port
Router(config-if)# shutdown
Router(config-if)# switchport access vlan <vlan_ID>
```

To disable a range of ports and place them into an unused VLAN:

```
Router(config)# interface range type slot/first-port - last-port
Router(config-if)# shutdown
Router(config-if)# switchport access vlan <vlan_ID>
```

APPENDIX A

Sample Configurations

These sample configurations are provided as general templates for initial configuration guidance. Each feature and command should be reviewed, tested and possibly revised according to the particular platform, software version and network architecture on which they are being deployed.

Sample TTY Ports Configuration

AUX Port

```
! AUX port access not required and access disabled
line aux 0
login
no password
!
```

Console Port

```
! Define local username for fallback access.
username adminuser privilege 15 password <mypassword>
!
aaa authentication login <authen-exec-list> group <adminAAAgroup> local-case
aaa authentication enable default group tacacs-group enable
aaa accounting exec <account-exec-list>  start-stop group <adminAAAgroup>
!
line con 0
 accounting exec <account-exec-list>
 login authentication <authen-exec-list>
!
! Local incoming access only
 transport input none
!
! Incoming access not permitted if the request does not specify the transport protocol
 transport preferred none
!
! No outgoing connections permitted
 transport output none
!
! Idle timeout of 3 minutes
 session-timeout 3
!
! EXEC timeout of 10 minutes
```

```
 exec-timeout 10 0
!
```

Sample VTY Lines Configuration

Sample Telnet Configuration

```
! Prevent hung sessions in case of a remote system crash
service tcp-keepalives-in
!
! Define local username for fallback access.
username adminuser privilege 15 password <mypassword>
!
aaa authentication login <authen-exec-list> group <adminAAAgroup> local-case
aaa authentication enable default group tacacs-group enable
aaa authorization exec <author-exec-list> group tacacs-group if-authenticated
aaa accounting exec <account-exec-list>  start-stop group <adminAAAgroup>

!
access-list <xACL#> remark ACL for Telnet
access-list <xACL#> permit tcp <NOCsubnet1> <inverse-mask> any eq 23
access-list <xACL#> permit tcp <NOCsubnet2> <inverse-mask> any eq 23
access-list <xACL#> deny ip any any log-input
!
```

Note that in access-class ACLs, destination should be any, and not a particular IP address of the router. If a specific host IP address is used, packets won’t match the ACE.

```
line vty 0 4
! Allow access from trusted hosts
 access-class <xACL#> in
!
! Incoming access via telnet only
 transport input telnet
!
 authorization exec <author-exec-list>
 accounting exec <account-exec-list>
 login authentication <authen-exec-list>
!
! No outgoing connections permitted
 transport output none
!
! Incoming access not permitted if the request does not specify the transport protocol
 transport preferred none
!
! Idle timeout of 3 minutes
 session-timeout 3
!
! EXEC timeout of 10 minutes
 exec-timeout 10 0
!
```

Sample SSH Configuration

```
! Prevent hung sessions in case of a remote system crash
service tcp-keepalives-in
!
! Define local username for fallback access.
username adminuser privilege 15 password <mypassword>
!
aaa authentication login <authen-exec-list> group <adminAAAgroup> local-case
aaa authentication enable default group tacacs-group enable
aaa authorization exec <author-exec-list> group tacacs-group if-authenticated
aaa accounting exec <account-exec-list>  start-stop group <adminAAAgroup>
!
access-list <xACL#> remark ACL for SSH
access-list <xACL#> permit tcp <NOCsubnet1> <inverse-mask> any eq 22
access-list <xACL#> permit tcp <NOCsubnet2> <inverse-mask> any eq 22
access-list <xACL#> deny ip any any log-input
!
crypto key generate rsa
! (after entering command, follow the series of prompts)
!
! SSH negotiation timeout of 30 seconds
ip ssh timeout 30
!
! SSH authentication attempts of 2 before an interface reset
ip ssh authentication-retries 2
!
line vty 0 4
 access-class <xACL#> in
!
! Incoming access via SSH only
 transport input ssh
!
 authorization exec <author-exec-list>
 accounting exec <account-exec-list>
 login authentication <authen-exec-list>
!
! No outgoing connections permitted
 transport output none
!
! Incoming access not permitted if the request does not specify the transport protocol
 transport preferred none
!
! Idle timeout of 3 minutes
 session-timeout 3
!
! EXEC timeout of 10 minutes
 exec-timeout 10 0
!
```

Sample Legal Banner Notification Configuration

```
! Present a legal notification banner approved by company legal counsel
banner login #
UNAUTHORIZED ACCESS TO THIS DEVICE IS PROHIBITED
You must have explicit, authorized permission to access or configure this device.
Unauthorized attempts and actions to access or use this system may result in civil and/or
criminal penalties.
All activities performed on this device are logged and monitored.
#
```

```
!
```

Sample AAA Services Configuration

```
! Enable AAA
aaa new-model
!
! Define the AAA servers
! If using TACACS+ servers, define the servers, the key to be used for each specific
server
!(rather than a global key) and enable the option to maintain a session to the server for
more
! efficient communication
! Note: The keys must match that entered for this device on the TACACS+ servers and should
follow
! general password policy guidelines (e.g. be at least 16 characters long)
!
tacacs-server host <TAC+server1> key <TAC+key1> single-connection
tacacs-server host <TAC+server2> key <TAC+key2> single-connection
!
! If using RADIUS servers, define the server, non-standard ports and the key to be used
for each ! server (rather than a global key).
! Note: The keys must match that entered for this device on the RADIUS servers and should
follow general password policy guidelines (e.g. be at least 16 characters long)
!
radius-server host <RADserver1> auth-port <port#> acct-port <port#> key <RADkey1>
radius-server host <RADserver2> auth-port <port#> acct-port <port#> key <RADkey1>
!
! Define a AAA group to be used to authenticate device access. Here the servers are
TACACS+
! servers
!
aaa group server tacacs+ <adminAAAgroup>
 server <TAC+server1>
 server <TAC+server2>
!
! This example shows how to define a group of RADIUS servers.
aaa group server radius <adminAAAgroup>
 server <RADIUSserver1>
 server <RADIUSserver2>
!
```

Note Be careful! This should apply to OOB interface. In addition, these ACLs have source and dest inversed.

```
! If using RADIUs, send the attribute ‘Service-Type’ in authentication requests to enable
the AAA server to only authorize device access to administrative users. This is
particularly important to implement if the same AAA server is being used for both
administrative and service users. The ‘Service-Type’ attribute is set to ‘1’ for login
access.
radius-server attribute 6 on-for-login-auth
!
! Enable support for RADIUS VSAs for both authentication and accounting
radius-server vsa send authentication
radius-server vsa send accounting
!
! Set the global default setting for the time (in seconds) to wait before re-attempting a
request, e.g. 1 second.
! Note: The timeout may be defined on an individual TACACS+ server basis as part of the
‘tacacs-server host’ definition. An individual server-defined setting takes priority over
the global setting.
```

```
tacacs-server timeout 5
!
! Note: The timeout may be defined on an individual RADIUS server basis as part of the
'radius-server host' definition. An individual server-defined setting takes priority over
the global setting.
radius-server timeout 5
!
! Set the global default setting for the number of times, e.g. 0, to re-attempt a response
from a particular RADIUS server before attempting communication with an alternative
server, if available. A setting of '0' will cause only one attempt to be made to each
server.
! Note: The retransmit time may be defined on an individual RADIUS server basis as part of
the 'radius-server host' definition. An individual server-defined setting takes priority
over the global setting.
radius-server retransmit 3
!

! Set the time (in minutes) to wait before re-attempting communication with a
non-responsive RADIUS server, e.g. 1 minute. This time only applies if multiple, redundant
RADIUS servers are defined.
radius-server deadtime 1
!
! Define the source interface to be used to communicate with the TACACS+ servers
ip tacacs source-interface <Loopback or OOB interface>
!
! Define the source interface to be used to communicate with the RADIUS servers
ip radius source-interface <Loopback or OOB interface>
!
! Define a AAA method list which enforces login authentication to an administrative server
group, with the local database defined as a fallback method in case of loss of
communication with the remote AAA servers
! Usernames and passwords with exec level access must be configured in the AAA server
and/or defined locally in order to obtain access
! Define local username for fallback access.
username adminuser privilege 15 password <mypassword>
!
! Note: Authentication attempts will NOT continue down a method list if a 'fail' response
is received, only if an error is received, e.g. if the server is down.
aaa authentication login <authen-exec-list> group <adminAAAgroup> local-case
!
! Define a AAA authorization method list for exec sessions to a AAA server, with local
fallback in case of loss of communication with the remote AAA servers

aaa authorization exec <author-exec-list> group <adminAAAgroup> if-authenticated
!
! Enable AAA accounting
aaa accounting exec default start-stop group <adminAAAgroup>p
aaa accounting commands 15 default start-stop group <adminAAAgroup>
aaa accounting system default start-stop group <adminAAAgroup>

!
! Ensure the session ID is maintained across all authentication, authorization and
accounting packets in a session
aaa session-id common
!
!
! Apply the authentication method to all active access lines to be secured, such as VTYs,
console, etc.
!
line [type]
 login authentication <authen-exec-list>
!
! Enforce exec session authorization on all active ports, for instance, VTY lines
! Ensure users are defined on the AAA server with the appropriate permissions
```

```
 authorization exec <author-exec-list>
!
!
```

Sample Web-Based GUI Configuration

Sample HTTP Configuration

```
! Disable HTTP access unless absolutely necessary
no ip http server
!
! Enable HTTP only if absolutely necessary
ip http server
!
! Require all access to be authenticated
! Note: Ensure the AAA server group and server configuration have been configured and at
least one username and password is available prior to applying this command. See the AAA
Configuration Guidelines section for complete details.
!
! Define local username for fallback access.
username adminuser privilege 15 password <mypassword>
!
aaa authentication login default group adminAAAgroup local-case
! Require AAA authorization for exec level commands
! Note: Ensure a AAA method list has been configured using ‘aaa authorization exec’ and at
least
! one username and password is available in order to enable access.
aaa authorization exec default group adminAAAgroup local
!
ip http authentication aaa
!
! Optionally enable accounting
aaa accounting exec default start-stop group adminAAAgroup
!
! Restrict where HTTP access attempts must originate from in order for them to be
processed,
! being as specific as possible, for instance, only permitting access from the NOC
subnets. HTTP
! access-class does not currently support the application of an extended ACL.
! The use of the log keyword enables unauthorized access attempts to be logged, thereby
enabling
! the tracking of abnormal activity. See the Logging Device Access section for more
information.
! Note: Ensure source IP spoofing protection mechanisms are also deployed
access-list <ACL#> remark ACL for HTTP
access-list <ACL#> permit <NOCsubnet1> <inverse-mask>
access-list <ACL#> permit <NOCsubnet2> <inverse-mask>
access-list <ACL#> deny any log
!
ip http access-class <ACL#>
!
! Restrict the maximum number of concurrent HTTP sessions, e.g. to 3.
ip http max-connections 3
!
! HTTP access requires telnet service to be enabled on VTY lines
line vty 0 4
 transport input telnet
```

Sample HTTPS Configuration

```
! Disable HTTP access
no ip http server
!
crypto key generate rsa
! (after entering command, follow the series of prompts)
!
! Enable HTTPS access
ip http secure-server
!
! Require all access to be authenticated
! Note: Ensure the AAA server group and server configuration have been configured and at
least one username and password is available prior to applying this command. See the AAA
Configuration Guidelines section for complete details.
!
! Define local username for fallback access.
username adminuser privilege 15 password <mypassword>
!
aaa authentication login default group adminAAAgroup local-case
! Require AAA authorization for exec level commands
! Note: Ensure a AAA method list has been configured using ‘aaa authorization exec’ and at
least
! one username and password is available in order to enable access.
aaa authorization exec default group adminAAAgroup local
!
ip http authentication aaa
!
! Optionally enable accounting
aaa accounting exec default start-stop group adminAAAgroup
!
!
! Restrict where HTTP access attempts must originate from in order for them to be
processed,
! being as specific as possible, for instance, only permitting access from the NOC
subnets. HTTP
! access-class does not currently support the application of an extended ACL.
! The use of the log keyword enables unauthorized access attempts to be logged, thereby
enabling
! the tracking of abnormal activity. See the Logging Device Access section for more
information.
! Note: Ensure source IP spoofing protection mechanisms are also deployed
access-list <ACL#> remark ACL for HTTP
access-list <ACL#> permit <NOCsubnet1> <inverse-mask>
access-list <ACL#> permit <NOCsubnet2> <inverse-mask>
access-list <ACL#> deny any log
!
ip http access-class <ACL#>
!
```

Sample SNMP Configuration

```
! Restrict where SNMP access attempts must originate from in order for them to be
processed,
! being as specific as possible, for instance, only permitting access from the SNMP
management
! stations.
access-list 55 remark ACL for SNMP access to device
access-list 55 permit 172.26.150.206
access-list 55 deny   any log
```

```
!
! Define an SNMP view which denies queries to download the full IP routing and ARP tables
snmp-server view myview internet included
snmp-server view myview ipRouteTable excluded
snmp-server view myview ipNetToMediaTable excluded
snmp-server view myview at excluded
!
! In case any of the above keywords are not recognized by a particular IOS release, use
the ones ! below
snmp-server view myview internet included
snmp-server view myview ip.21 excluded
snmp-server view myview ip.22 excluded
snmp-server view myview mib-2.3 excluded
!
! Once the views are defined, associate them to the SNMP community strings. In addition,
restrict ! access to those communities to read-only (RO) and apply the previously defined
ACL.
snmp-server community mycommunity view myview  RO 55
!
! Define the SNMP managers, their supported SNMP version and permitted community string.
snmp-server engineID remote 172.26.150.206 80000009030000B064EFE100
snmp-server host 172.26.150.206 version 3 priv mycommunity
!
!If SNMP v3 is supported, define an SNMP v3 user group with security level ‘authPriv’ and
enforce ! a restricted view
snmp-server group mygroup v3 priv read myview
!
!Define a user within the SNMP v3 group, along with their authentication password and
encryption
!key
snmp-server user admin mygroup v3 encrypted auth md5 cisco123 priv des56 mykey
!
! Set the source IP address for SNMP traps to the address used on the administrative
loopback
! interface of OOB interface.
snmp-server trap-source GigabitEthernet1/2
!
! Configure the system to send a trap on SNMP authentication failure.
snmp-server enable traps snmp authentication
!
!Configure the System to send a trap for configuration changes.
snmp-server enable traps config
!
!Configure the system to send a trap for environmental monitor threshold exceptions
snmp-server enable traps envmon
!
!Enable any additional required traps. It is recommended that only operationally important
traps
!are enabled, e.g. BGP state changes. Ensure enabled traps are monitored.
snmp-server enable traps bgp
!
!If configuration files are downloaded via SNMP by TFTP servers, restrict which TFTP
servers may
!do so.
snmp-server tftp-server-list 55!
```

Sample Timestamps and NTP Configuration

The following are the configuration fragments for the WAN edge and branch routers used in our validation lab. In this scenario, the WAN edge routers were configured as time servers, and the branch routers as clients. The WAN edge routers are synchronized with an internal time server accessible throughout an Out of Band management network. Communication between branch routers and the WAN edge routers is inband (uses the data network).

NTP Server Configured as Master Stratus 3

```
! Enables timestamp information for debug messages
service timestamps debug datetime localtime show-timezone msec
!
! Enables timestamp information for log messages
service timestamps log datetime localtime show-timezone msec
!
! This router has a hardware calendar that is used as an
! authoritative time source
clock calendar-valid
!
! This allows NTP to update the hardware calendar chip.
ntp update-calendar
!

! Sets the network-wide zone to Eastern Standard Time (UTC -5 hours)
clock timezone EST -5
!
! Defines summertime adjustment for Eastern Daylight Saving Time (UTC -4 hours)
clock summer-time EDT recurring
!
! Sets source of NTP packets to the IP address used on the OOB interface
ntp source GigabitEthernet1/2
!
! Sets the router as a NTP master clock to which peers may synchronize in case
! an external NTP source is not available.
ntp master 3
!
! Defines server and peer
ntp server 172.26.129.252
ntp peer 172.26.159.163
!
! Enables authentication
ntp authentication-key 10 md5 00071A1507545A545C 7
ntp trusted-key 10
ntp authenticate
!
! Enforces a list of allowed NTP servers and peers
access-list 10 permit 127.127.7.1
access-list 10 permit 172.26.129.252
access-list 10 permit 172.26.159.163
access-list 10 deny any log
!
ntp access-group peer 10
!
! Enforces what clients may use this server
access-list 20 permit 10.139.5.11
access-list 20 permit 10.139.5.9
access-list 20 permit 10.139.5.7
```

```
access-list 20 deny any log
!
ntp access-group serve-only 20
```

Example NTP Client (Stratus 4)

```
! Enables timestamp information for debug messages
service timestamps debug datetime localtime show-timezone msec
!
! Enables timestamp information for log messages
service timestamps log datetime localtime show-timezone msec
!
! Sets the network-wide zone to Eastern Standard Time (UTC -5 hours)
clock timezone EST -5
!
! Defines summertime adjustment for Eastern Daylight Saving Time (UTC -4 hours)
clock summer-time EDT recurring
!
! Defines server
ntp server 172.26.158.236
!
! Enables authentication
ntp authentication-key 10 md5 00071A1507545A545C 7
ntp trusted-key 10
ntp authenticate!
```

Sample Syslog Configuration

Note Ensure timestamps and NTP are enabled on a device prior to enabling syslog.

```
! Disable console logging
no logging console
!
! Define the source IP address for syslog messages
logging source-interface Loopback0
!
! Define the syslog servers
logging <syslog-server1>
logging <syslog-server2>
!
! Enable a local history buffer log, e.g. of 64k bytes, for debug level messages
logging buffered 64000 debugging
!
! Define a facility to be used by the syslog server to determine where to store the traps
logging facility <facility>
!
! Enable ‘informational’ level syslog traps to be sent
logging trap informational
!
! Enable syslog rate-limiting for levels 3 and above
logging rate-limit <#msgs/sec> except 2
!
```

Disabling Unnecessary Services

```
! BOOTP, IP Source Routing, PAD Disable global service on ALL ROUTERS
no ip bootp server
no ip source-route
no service pad


! Global Services disabled by default (all routers)
no service finger
no ip identd
no service tcp-small-servers
no service udp-small-servers

! Disable CDP, MOP, IP Redirects on EXTERNAL facing interfaces
interface <interface-type/number>
 no cdp enable
 no mop enabled
 no ip redirects
 no ip proxy-arp

! Disable MOP, IP Redirects on ACCESS interfaces
interface <interface-type/number>
 no mop enabled
 no ip redirects
 no ip proxy-arp

! Interface services disabled by default
interface <interface-type/number>
 no ip directed-broadcast
```

Sample iACL Configurations

iACL at Internet Edge

The example below shows an iACL protecting an enterprise Internet Edge, and involving the following:

- The enterprise is assigned the 198.133.219.0/24 address block
- The enterprise edge router (198.133.219.6) has a BGP peering session with 198.133.219.10

The iACL shown below was developed based on this information. The ACL permits external BGP peering to the external peer, provides anti-spoof filters, and protects the infrastructure from all external access.

```
!
! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!--- Module 1: Anti-spoofing Denies
!--- These ACEs deny fragments, RFC 1918 space,
!--- invalid source addresses, and spoofs of
!--- internal space (space as an external source).
!
!--- Deny fragments.
access-list 110 deny tcp any 198.133.219.0 0.0.0.255 fragments
access-list 110 deny udp any 198.133.219.0 0.0.0.255 fragments
access-list 110 deny icmp any 198.133.219.0 0.0.0.255 fragments
!--- Deny special-use address sources.
!--- See RFC 3330 for additional special-use addresses.
```

```
access-list 110 deny ip host 0.0.0.0 any
access-list 110 deny ip 127.0.0.0 0.255.255.255 any
access-list 110 deny ip 192.0.2.0 0.0.0.255 any
access-list 110 deny ip 224.0.0.0 31.255.255.255 any
!--- Filter RFC 1918 space.
access-list 110 deny ip 10.0.0.0 0.255.255.255 any
access-list 110 deny ip 172.16.0.0 0.15.255.255 any
access-list 110 deny ip 192.168.0.0 0.0.255.255 any
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!--- Module 2:  Explicit Permit
!--- Permit only applications/protocols whose destination
!--- address is part of the infrastructure IP block.
!--- The source of the traffic should be known and authorized.
!
!--- Note: This template must be tuned to the network’s
!--- specific source address environment. Variables in
!--- the template need to be changed.
!--- Permit external BGP.
access-list 110 permit tcp host 198.133.219.10 host 198.133.219.6 eq bgp
access-list 110 permit tcp host 198.133.219.10 eq bgp host 198.133.219.6
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!--- Module 3:  Explicit Deny to Protect Infrastructure
access-list 110 deny ip any 198.133.219.0 0.0.0.255
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!--- Module 4:  Explicit Permit for Transit Traffic
access-list 110 permit ip any any
```

iACL at WAN Edge

This example corresponds to an enterprise WAN edge. The objective of the iACL is to protect the core infrastructure from threats rising from the branches. This scenario involves the following:

172.16.0.0/16 is reserved to OBB network. No packets in this range should come from the branches.

10.122.0.0/16 is allocated to the core infrastructure devices. No packets in this range should come from the branches.

10.139.5.0/24 is allocated to the WAN links. Branch routers are the only systems expected to send packets from this network range, and for the following purposes:

- -EIGRP routing
- ICMP
- SSH (client and server)
- TACACS+
- NTP
- Syslog

```
!--- Module 1: Anti-spoofing, deny special use addresses
! Deny your OOB address space as a source in packets
access-list 101 deny ip 172.26.0.0 0.0.255.255 any
! Deny src addresses of 0.0.0.0 and 127/8 (special use IPv4 addresses RFC3330)
access-list 101 deny ip host 0.0.0.0 any
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
access-list 101 deny ip 192.0.2.0 0.0.0.255 any
```

```
access-list 101 deny ip 224.0.0.0 31.255.255.255 any
!
!--- Module 2:  Explicit Permit
! Permit traffic from routing peer
access-list 101 permit eigrp host 10.139.5.11 host 224.0.0.10
access-list 101 permit icmp host 10.139.5.11 172.26.0.0 0.0.255.255
access-list 101 permit tcp host 10.139.5.11 eq 22 172.26.0.0 0.0.255.255
access-list 101 permit tcp host 10.139.5.11 172.26.0.0 0.0.255.255 eq 22
access-list 101 permit tcp host 10.139.5.11 host 172.26.150.206 eq 49
access-list 101 permit udp host 10.139.5.11 eq ntp host 172.26.158.236 eq ntp
access-list 101 permit udp host 10.139.5.11 host 172.26.150.206 eq syslog
!
!--- Module 3:  Explicit Deny to Protect Infrastructure
! Deny all other access to infrastructure
access-list 101 deny ip any 10.139.5.0 0.0.0.255
access-list 101 deny ip any 10.122.0.0 0.0.255.255
access-list 101 deny ip any 172.26.0.0 0.0.255.255
!
!--- Module 4:  Explicit Permit/Deny for Transit Traffic
! Permit transit traffic enterprise inner iACL
access-list 101 permit ip any any
```

Sample rACL Configurations

The following is an example rACL protecting an enterprise edge router in a scenario involving the following addresses:

- Public address block is 198.133.219.0/24
- Public infrastructure block is 198.133.219.0/28
- External routing IP address is 198.133.219.5/32
- Out of band management segment is 172.26.0.0/16, router address is 172.26.159.164
- Private address space is 10.135.5.0/24 (directly connected to router)

Given this information, the required rACL could be something like the example shown below. This sample rACL starts with the necessary deny statements to block fragments, then continues with a list of explicit permit statements that allow the expected management and controls protocols, such as BGP, OSPF, SNMP, and NTP. Finally, the rACL ends with a explicit deny entry to block any unexpected traffic sent to the RP.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Module 1 - Deny fragments Section
! Denies any non-initial fragments directed to the RP
access-list 110 deny tcp any any fragments
access-list 110 deny udp any any fragments
access-list 110 deny icmp any any fragments
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Module 2 - Explicit Permit Section
!
! Explicit Permit Phase. Permit only applications whose
! destination address are the router's valid infrastructure
! addresses and that come from an valid host.
!
! Permit BGP session (outside)
access-list 110 permit tcp 198.133.219.0 0.0.0.15 host 198.133.219.5 eq bgp
access-list 110 permit tcp 198.133.219.0 0.0.0.15 eq bgp host 198.133.219.5
!
```

```
! Permit OSPF (inside)
access-list 110 permit ospf 10.135.5.0 0.0.0.255 host 224.0.0.5
!
! DR multicast address, if needed (inside)
access-list 110 permit ospf 10.135.5.0 0.0.0.255 host 224.0.0.6
access-list 110 permit ospf 10.135.5.0 0.0.0.255 host 169.223.253.13
access-list 110 permit ospf 10.135.5.0 0.0.0.255 host 169.223.253.15
access-list 110 permit ospf 10.135.5.0 0.0.0.255 host 169.223.253.19
!
! permit EIGRP (inside)
access-list 110 permit eigrp 10.135.5.0 0.0.0.255 host 224.0.0.10
access-list 110 permit eigrp 10.135.5.0 0.0.0.255 host 169.223.253.13
access-list 110 permit eigrp 10.135.5.0 0.0.0.255 host 169.223.253.15
access-list 110 permit eigrp 10.135.5.0 0.0.0.255 host 169.223.253.19
!
! Remote access: ssh (out of band)
access-list 110 permit tcp 172.26.0.0 0.0.255.255 host 172.26.159.164 eq 22
!
! SNMP(out of band)
access-list 110 permit udp 172.26.0.0 0.0.255.255 host 172.26.159.164 eq snmp
!
! NTP (out of band)
access-list 110 permit udp 172.26.0.0 0.0.255.255 host 172.26.159.164 eq ntp
!
! Router originated traceroute
! Each hop returns a ttl exceeded (type 11, code 3) message
! and the final destination returns an ICMP port unreachable
! (type 3, code 0)
access-list 110 permit icmp any host 198.133.219.5 ttl-exceeded
access-list 110 permit icmp any host 198.133.219.5 port-unreachable
access-list 110 permit icmp any host 169.223.253.13 ttl-exceeded
access-list 110 permit icmp any host 169.223.253.13 port-unreachable
access-list 110 permit icmp any host 169.223.253.15 ttl-exceeded
access-list 110 permit icmp any host 169.223.253.15 port-unreachable
access-list 110 permit icmp any host 169.223.253.19 ttl-exceeded
access-list 110 permit icmp any host 169.223.253.19 port-unreachable
!
! TACACS+ for router authentication (out of band)
access-list 110 permit tcp 172.26.0.0 0.0.255.255 host 172.26.159.164 established
!
! RADIUS (out of band)
access-list 110 permit udp 172.26.0.0 0.0.255.255 host 172.26.159.164
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Module 2 – Explicit denies all other traffic
!
! Deny all other traffic
access-list 110 deny ip any any
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Activation of rACL
!
 ip receive access-list 110
```

CoPP Sample Configuration

The following example shows how to develop a CoPP policy and how to apply it in order to protect the control plane of an Internet Edge router.

The following assumption are made:

- Public address block is 198.133.219.0/24
- The public infrastructure block is 198.133.219.0/28
- The external routing IP address is 198.133.219.5/32
- Out of band management segment is 172.26.0.0/16, router IP is 172.26.159.164
- Private address space is 10.135.5.0/24 (directly connected to router)

In this example, the control plane traffic is classified based on relative importance and traffic type. Nine classes are defined, each of which is associated with a separate extended ACL:

- BGP (coppacl-bgp): BGP traffic
- IGP (coppacl-igp): OSPF traffic
- Interactive Management (coppacl-interactivemanagement): remote access and management traffic such as TACACS, SSH, SNMP, and NTP.
- File Management (coppacl-filemanagement): remote file transfer traffic such as TFTP and FTP.
- Reporting (coppacl-reporting): SAA generated ICMP requests from SAA source routers
- Monitoring (coppacl-monitoring): ICMP and traceroute traffic
- Critical Applications (coppacl-critical-app): HSRP traffic
- Undesirable Traffic (coppacl-undesirable): explicitly denies unwanted traffic (for example, Slammer worm packets)
- Default (no ACL needed): all traffic received by the control plane that has not been otherwise identified.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Sample ACLs for CoPP classification
!
! Class for BGP traffic
ip access-list extended coppacl-bgp
 remark BGP traffic class
! allow BGP from routers in the infrastructure block to this router's BGP TCP port
 permit tcp 198.133.219.0 0.0.0.15 host 198.133.219.5 eq bgp
 permit tcp 198.133.219.0 0.0.0.15 eq bgp host 198.133.219.5
!
! Permit OSPF sessions to peers on the local 10.135.5.0/24 network
ip access-list extended coppacl-igp
 remark IGP traffic class
 permit ospf 10.135.5.0 0.0.0.255 host 224.0.0.5
 permit ospf 10.135.5.0 0.0.0.255 host 224.0.0.6
 permit ospf 10.135.5.0 0.0.0.255 host 169.223.253.13
 permit ospf 10.135.5.0 0.0.0.255 host 169.223.253.15
 permit ospf 10.135.5.0 0.0.0.255 host 169.223.253.19
!
! The Interactive Management class is for traffic that is required for accessing
! and managing the router, in this example, TACACS, ssh,
! snmp, and ntp is classified in this class. Interactive traffic is expected
! to be originated from the Out of band network (172.26.0.0/16)
```

```
ip access-list extended coppacl-interactivemanagement
 remark CoPP interactive management traffic class
! permit return traffic from TACACS+ Servers
 permit tcp 172.26.0.0 0.0.255.255 host 172.26.159.164 established
! SSH access to the router from a subnet
 permit tcp 172.26.0.0 0.0.255.255 host 172.26.159.164 eq 22
! SNMP access from the management segment to the router
 permit udp 172.26.0.0 0.0.255.255 host 172.26.159.164 eq snmp
! Allow the router to receive NTP packets from a known clock sources
 permit udp 172.26.0.0 0.0.255.255 host 172.26.159.164 eq ntp
!
! The File Management class is for file transfer traffic required for software
! and configuration maintenance, in this example, TFTP and FTP is classified in this class
ip access-list extended coppacl-filemanagement
 remark CoPP file management traffic class
! Allow router initiated FTP (active and passive)
 permit tcp 172.26.0.0 0.0.255.255 eq 21 host 172.26.159.164 gt 1023 established
 permit tcp 172.26.0.0 0.0.255.255 eq 20 host 172.26.159.164 gt 1023
 permit tcp 172.26.0.0 0.0.255.255 gt 1023 host 172.26.159.164 gt 1023 established
! Allow router initiated TFTP
 permit udp 172.26.0.0 0.0.255.255 gt 1023 host 172.26.159.164 gt 1023
!
! The reporting class is for traffic used for generating network
! performance statistics. In this example, we are using SAA to
! generate ICMP Pings with different DSCP bits in order to determine
! response times for classes of traffic.  i.e. COS1 will use an ICMP
! with DSCP set to EF, COS2=AF31, COS3=AF21 and COS4=BE.  We will
! create an ACL to classify ICMP pings from specific source routers
! using SAA to generate the ICMPs. We will then use this ACL and the
! ‘match ip dscp’ classification criteria in the service policy to
! create the reporting class.
ip access-list extended coppacl-reporting
 remark CoPP reporting traffic class
! permit SAA generated ICMP requests from SAA source routers
 permit icmp 172.26.0.0 0.0.255.255 host 172.26.159.164 echo
!
! The monitoring class is used for traffic that is required for
! monitoring the router. Monitoring traffic is traffic that we expect
! to see destined to the router and want to track and limit
ip access-list extended coppacl-monitoring
 remark CoPP monitoring traffic class
! permit router originated traceroute
 permit icmp any host 198.133.219.5 ttl-exceeded
 permit icmp any host 198.133.219.5 port-unreachable
 permit icmp any host 169.223.253.13 ttl-exceeded
 permit icmp any host 169.223.253.13 port-unreachable
 permit icmp any host 169.223.253.15 ttl-exceeded
 permit icmp any host 169.223.253.15 port-unreachable
 permit icmp any host 169.223.253.19 ttl-exceeded
 permit icmp any host 169.223.253.19 port-unreachable
! permit receipt of responses to router originated pings
 permit icmp any any echo-reply
! allow pings to router
 permit icmp any any echo
!
! The critical-app class is used for traffic that is crucial to
! particular customer’s environment. In this example, HSRP.
ip access-list extended coppacl-critical-app
 remark CoPP critical apps traffic class
! permit HSRP
 permit ip any host 224.0.0.2
```

```
!
! This ACL identifies traffic that should always be blocked from
! accessing the Route Processor.! Once undesirable traffic flow is
! identified, an access list entry classifying it can be added and mapped to the
! undesirable traffic class. This can be used as a reaction tool.
ip access-list extended coppacl-undesirable
 remark explicitly defined "undesirable" traffic
! permit, for policing, all traffic destined to UDP 1434
 permit udp any any eq 1434
```

Table A-1 Sample CoPP Policy

Traffic class	Rate (bps)	Conform action	Exceed action
BGP	80,000	Transmit	Drop
IGP	N/A	Transmit	Transmit
Interactive Management	10,000,000	Transmit	Drop
File Management	N/A	Transmit	Transmit
Reporting	500,000	Transmit	Drop
Monitoring	500,000	Transmit	Drop
Critical Apps	500,000	Transmit	Drop
Undesirable	50,000	Drop	Drop
Default	10,000,000	Transmit	Drop

Once the control plane traffic has been classified, the next step is to define the policy action for each traffic class. In this example the limits set per each class represent the boundary after which the system becomes unresponsive and starts dropping packets. Our intention is to deploy a policy that protects the router while reducing the risk of dropping critical traffic. To that end, CoPP policies are configured to permit each traffic class with an appropriate rate limit. Table A-1 shows the parameters used in the CoPP policies.

Note The rates defined in Table A-1 were successfully tested on a Cisco 7200 VXR Series Router with NPE-G1. It is important to note that the values here presented are solely for illustration purposes; every environment will have different baselines.

The following is the policy for the configuration described inTable A-1:

```
! Define a class for each "type" of traffic and associate it with an ACL
class-map match-all coppclass-bgp
  match access-group name coppacl-bgp
class-map match-all coppclass-igp
  match access-group name coppacl-igp
class-map match-all coppclass-interactivemanagement
  match access-group name coppacl-interactivemanagement
class-map match-all coppclass-filemanagement
  match access-group name coppacl-filemanagement
class-map match-all coppclass-reporting
  match access-group name coppacl-reporting
  match ip dscp ef af31 af21 default
class-map match-all coppclass-monitoring
```

```
  match access-group name coppacl-monitoring
class-map match-all coppclass-critical-app
  match access-group name coppacl-critical-app
class-map match-all coppclass-undesirable
  match access-group name coppacl-undesirable
!
! This is the actual policy. Depending on class of traffic, rates and associated actions
! are defined
policy-map copp-policy
!
! BGP traffic is limited to a rate of 80,000 bps, if traffic exceeds
! that rate it is dropped. NOTE: In this example BGP traffic is rate-limited
! to control attacks based on BGP packets. Once the normal rates are determined,
! and depending on the hardware platform used, it’s recommended you consider
! readjusting the rate-limiting parameters.
  class coppclass-bgp
     police 80000 8000 8000 conform-action transmit exceed-action drop
!
! IGP traffic will not be limited in this example either therefore no
! operation needs to be specified in this class. NOTE: As with the BGP
! class, once normal rates are determined for your IGP traffic, you may
! consider setting a rate-limit to further protect your router.
  class coppclass-igp
!
! Interactive Management traffic is limited to a rate of 10,000,000 bps,
! if traffic exceeds that rate it is dropped.
  class coppclass-interactivemanagement
     police 10000000 100000 100000 conform-action transmit exceed-action drop
!
! File Management traffic will not be limited in this example either therefore no
! operation needs to be specified in this class. NOTE: As with the IGP
! class, once normal rates are determined for your file management traffic,
! you may consider setting a rate-limit to further protect your router.
  class coppclass-filemanagement
!
! Reporting traffic is limited to a rate of 500,000 bps, if traffic exceeds
! that rate it is dropped
  class coppclass-reporting
     police 500000 5000 5000 conform-action transmit exceed-action drop
!
! Monitoring traffic is limited to a rate of 500,000 bps, if traffic exceeds
! that rate it is dropped
  class coppclass-monitoring
     police 500000 5000 5000 conform-action transmit exceed-action drop
!
! critical-app traffic is limited to a rate of 500,000 bps, if traffic
! exceeds that rate it is dropped
  class coppclass-critical-app
     police 500000 5000 5000 conform-action transmit exceed-action drop
!
! This policy drops all traffic categorized as undesirable, regardless
! of rate.
  class coppclass-undesirable
     police 50000 1500 1500 conform-action drop exceed-action drop
! or if on the T train you can use the drop command
!  drop
!
! The default class applies to all traffic received by the control
! plane that has not been otherwise identified. In this example, all
! default traffic is limited to 10,000,000 bps and violations of that limit
! are dropped.
```

```
  class class-default
     police 10000000 100000 100000 conform-action transmit exceed-action drop
….
! Applies the defined CoPP policy to the control plane
Router(config)# control-plane
Router(config-cp)# service-policy input copp-policy
```

Control Plane Protection Sample Configuration

Assuming that a control plane protection has been configured previously using MQC CLI, the following example shows how the policy is applied to the control-plane host subinterface:

```
Router(config)# control-plane host
Router(config-cp)# service-policy input copp-policy
```

The following example shows how to configure a port-filter policy to drop all traffic destined to closed or "nonlistened" TCP/UDP ports:

```
Router(config)# class-map type port-filter pf-class
Router(config-cmap)# match closed-ports
Router(config-cmap)# exit
Router(config)# policy-map type port-filter pf-policy
Router(config-pmap)# class pf-class
Router(config-pmap-c)# drop
Router(config-pmap-c)# end
Router#
```

The following example shows how to configure a queue-threshold policy to set the queue limit for SNMP protocol traffic to 50, Telnet traffic to 50, and all other protocols to 150:

```
Router(config)# class-map type queue-threshold qt-snmp-class
Router(config-cmap)# match protocol snmp
Router(config-cmap)# class-map type queue-threshold qt-telnet-class
Router(config-cmap)# match protocol telnet
Router(config-cmap)# class-map type queue-threshold qt-other-class
Router(config-cmap)# match host-protocols
Router(config-cmap)# exit
Router(config)# policy-map type queue-threshold qt-policy
Router(config-pmap)# class qt-snmp-class
Router(config-pmap-c)# queue-limit 50
Router(config-pmap-c)# class qt-telnet-class
Router(config-pmap-c)# queue-limit 50
Router(config-pmap-c)# class qt-other-class
Router(config-pmap-c)# queue-limit 150
Router(config-pmap-c)# end
Router#
```

APPENDIX B

Commonly Used Protocols in the Infrastructure

Table B-1 lists commonly used protocols found in the infrastructure.

Table B-1 Commonly Used Protocols in the Infrastructure

Protocol	Protocol Number, TCP/UDP Ports, Message Type
BGP	TCP/179
OSPF	Prot 89
EIGRP	Prot 88
GRE	Prot 47
AH	Prot 51
ESP	Prot 50
TACACS+	TCP/49
RADIUS	UDP/1812, UDP/1813; in the past UDP/1645 and UDP/1646
SSH	TCP/22
TELNET	TCP/23
SNMP	UDP/161
NTP	UDP/123
ICMP	Prot 1, ttl-exceeded, port-unreachable, echo, echo-reply
DNS	UDP/53

APPENDIX C

Related Documents

- Infrastructure Protection Control Plane Security Overview in Cisco IOS Software

  http://www.cisco.com/en/US/products/ps6642/products_white_paper0900aecd805ffde8.shtml
- Deploying Control Plane Policing

  http://www.cisco.com/en/US/products/ps6642/products_white_paper0900aecd804fa16a.shtml
- Control Plane Protection

  http://www.cisco.com/en/US/products/ps6441/products_feature_guide09186a0080556710.html
- Control Plane Logging

  http://www.cisco.com/en/US/products/ps6441/products_feature_guide09186a00806107d6.html
- Infrastructure Protection on Cisco IOS Software-based Platforms

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_white_paper0900aecd802b8f21.shtml

  http://www.cisco.com/en/US/prod/collateral/iosswrel/ps8802/ps6970/ps1838/prod_white_paper0900aecd804ac831.pdf
- Management Plane Protection

  http://www.cisco.com/en/US/products/ps6441/products_feature_guide09186a0080617022.html
- Memory Threshold Notifications

  http://www.cisco.com/en/US/products/ps6642/products_white_paper09186a00801cd891.shtml
- Infrastructure Protection on Cisco Catalyst 6500 and 4500 Series Switches

  http://www.cisco.com/application/pdf/en/us/guest/netsol/ns171/c649/ccmigration_09186a0080825564.pdf

APPENDIX D

Infrastructure Device Access Checklist

Feature	Task	Task Completed?	Comments/Notes
Restrict Infrastructure Device Accessibility	Review all available terminal and management ports and services		
	Disable all terminal and management ports that are not explicitly required or actively being used		
	Only permit device access through required and supported services and protocols, using only secure access protocols such as SSH and HTTPS where possible		
	Only accept access attempts to authorized ports and services from authorized originators		
	Deny unused and unnecessary terminal and management services and protocols, e.g. telnet, HTTP		
	Deny outgoing access unless explicitly required		
	Authenticate all terminal and management access using centralized (or local) AAA		
	Authenticate all EXEC level terminal and management access using centralized (or local) AAA		
	Authorize all interactive and privileged EXEC level device management access using centralized (or local) AAA		

Enforce Session Management	Enforce an idle timeout to detect and close inactive sessions		
	Enforce an active session timeout to restrict the maximum duration of a session prior to re-authentication		
	Detect and close hung sessions, e.g. using keepalives		
Restrict Device Access Vulnerability to Dictionary and DoS Attacks	Enforce a strong password policy (may be done on the AAA server)		
	Restrict the frequency of login attempts		
	Enforce a lockout period upon multiple authentication failure attempts within a defined time window (may be done on the AAA server)		
	Restrict the maximum number of concurrent sessions		
	Reserve one terminal or management port for access solely by one particular NoC host		
Legal Notification	Present legal notification banner upon all terminal, management and privileged EXEC level access		
AAA Server Communication Security	Employ strong secrets for authentication between the AAA server and NAS		
	Restrict AAA communication to only the limited set of authorized AAA servers, and over the configured AAA communication ports		

Web-based GUI Access	Disable HTTP/HTTPS access if not required		
	Only permit web access from authorized originators		
	Restrict access to HTTPS only if web access required		
	Authenticate and authorize all web access using centralized (or local) AAA		
	Authorize all web access using centralized (or local) AAA		
	Enforce an idle timeout to detect and close inactive sessions		
	Enforce an active session timeout to restrict the maximum duration of a session prior to re-authentication		
	Detect and close hung sessions, e.g. using keepalives		
	Restrict the permitted rate of login attempts		
	Restrict the maximum number of concurrent sessions		
SNMP Access	Disable SNMP access if not required		
	Only use SNMP v3 where possible		
	Delete default community strings		
	Only permit SNMP access from authorized originators		
	Only enable minimum required access, e.g. read-only		
	Define strong, non-trivial community strings where SNMP required		
	Restrict SNMP views per community where possible		
	Enable only operationally important traps		
	Block queries that may impact device performance		
Locally Stored Information Protection	Enforce strong encryption of locally stored information		

Infrastructure Device Management Access Logging	Configure NTP across all devices (see NTP section for details)		
	Log all successful interactive device management access using centralized AAA or an alternative, e.g. syslog		
	Log all successful privileged EXEC level device management access using centralized AAA or an alternative, e.g. syslog		
	Log all failed interactive device management access using centralized AAA or an alternative, e.g. syslog		
	Log all failed privileged EXEC level device management access using centralized AAA or an alternative, e.g. syslog		
	Log all commands entered at a privileged EXEC level using centralized AAA or an alternative		
	Send an SNMP trap on community name authentication failures to track failed access attempts		
	Send an SNMP trap for configuration changes and environmental monitor threshold exceptions		
	Log all system-level events, e.g. reboot, accounting on/off, using centralized AAA or an alternative		
Secure File Management	Permit only secure file transfer, e.g. SCP, where possible		
	Block insecure file transfer, e.g. FTP, TFTP, unless required		
	Device software image verification, e.g. MD5		
Device Management Best Common Practices	Assign unique, per-user accounts		
	Remove default accounts and passwords		
	Force users to periodically change their password		
	Use TACACS+ for administrative device access where possible		
	Define multiple servers for redundancy, e.g. AAA, NTP, syslog, SNMP		
	Only grant minimum access privileges		
	Review the password recovery settings		