CHAPTER 2

Network Foundation Protection

This chapter describes the best practices for securing the network infrastructure itself. This includes setting a security baseline for protecting the control and management planes as well as setting a strong foundation on which more advanced methods and techniques can subsequently be built on. Later in this chapter, each design module is presented with the additional security design elements required to enhance visibility and control and to secure the data plane.

The following are the key areas of baseline security:

- Infrastructure device access
- Routing infrastructure
- Device resiliency and survivability
- Network telemetry
- Network policy enforcement
- Switching infrastructure

For more detailed information on deployment steps and configurations, refer to the *Network Security Baseline* document at the following URL:

http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/Baseline_Security/securebasebook.html

Key Threats in the Infrastructure

The following are some of the expected threats to the network infrastructure:

- Denial-of-service (DoS)
- Distributed DoS (DDoS)
- Unauthorized access
- Session hijacking
- Man-in-the-middle (MITM) attack
- Privilege escalation
- Intrusions
- Botnets
- Routing protocol attacks
- Spanning tree attacks

- Layer 2 attacks

Infrastructure Device Access Best Practices

Securing the network infrastructure requires securing the management access to these infrastructure devices. If the 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 measures to securing both interactive and management access to an infrastructure device are as follows:

- *Restrict device accessibility*—Limit the accessible ports and restrict the permitted communicators and 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.

Protect Local Passwords

Passwords should generally be maintained and controlled by a centralized AAA server. However, the Cisco IOS and other infrastructure devices generally store some sensitive information locally. Some local passwords and secret information may be required such as for local fallback in the case of AAA servers not being available, special-use usernames, secret keys, and other password information.

Global password encryption, local user-password encryption, and enable secret are features available in the Cisco IOS to help secure locally stored sensitive information:

- Enable automatic password encryption with the **service password-encryption** global command. Once configured, all passwords are encrypted automatically, including passwords of locally defined users.
- Define a local enable password using the **enable secret** global command. Enable access should be handled with an AAA protocol such as TACACS+. The locally configured enable password will be used as a fallback mechanism after AAA is configured.

- Define a line password with the **password** line command 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.

Note that the encryption algorithm used by the service **password-encryption** command is a Vigenere cipher (Type 7) that can be easily reversed. Consequently, this command is primarily useful for keeping unauthorized individuals from viewing passwords in the configuration file simply by looking over the shoulder of an authorized user.

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.

The following configuration fragment illustrates the use of the recommended commands:

```
service password-encryption
enable secret <strong-password>
  line vty 0 4
  password <strong-password>
```

Implement 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 that may be included in a legal notification banner include the following:

- 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 standpoint, rather than a legal, 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.

The following example displays the banner after the user logs in:

```
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.
#
```

Note In Cisco IOS, a number of banner options are available, including **banner motd**, **banner login**, **banner incoming**, and **banner exec**. For more information on these commands, refer to the Cisco IOS Command Reference on cisco.com.

Enforce Authentication, Authorization and Accounting (AAA)

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

- *Authentication*—Enables users 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. All management access (SSH, Telnet, HTTP, and HTTPS) should be controlled with AAA.

Due to the fact that RADIUS does not support command authorization, the protocol is not as useful as TACACS+ when it comes to device administration. TACACS+ supports command authorization, allowing the control of which command can be executed on a device and which cannot. For this reason, this guide focuses on TACACS+ and not on RADIUS. For information on how to configure RADIUS for device management, refer to the *Network Security Baseline* or the Cisco IOS user documentation on cisco.com.

The following are the best practices for enabling TACACS+ on Cisco IOS:

- Enable AAA with the **aaa new-model** global command. Configure the **aaa session-id common** command to ensure the session ID is maintained across all AAA packets in a session.
- Define server groups of all AAA servers. If possible, use a separate key per server. Set source IP address for TACACS+ communications, preferably use the IP address of a loopback or the out-of-band (OOB) management interface.
- Define a login authentication method list and apply it to console, VTY, and all used access lines. Use TACACS+ as the primary method and local authentication as fallback. Do not forget to define a local user for local fallback.
- Authenticate enable access with TACACS+, and use local enable as fallback method. Configure a TACACS+ enable password per user.
- 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. 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+, configure the user or group profile with the “privilege level” attribute to 15.

- Enforce console authorization: By default, authorization on the console port is not enforced. It is a good practice to enable console authorization with the **aaa authorization console** command to ensure access is granted only to users with an administrative access privilege.
- Enable command authorization for privilege level 15: By default, administrative access to IOS has a privilege level 15. Enable the **command authorization** command for the privilege level 15 and any other if defined.
- Activate the **exec accounting** command to monitor shell connections. Enable the **accounting** command for the privilege levels to be used. Activate system accounting for system-level events.

Note Enable access can be automatically granted as a result of exec authorization. To that end, TACACS+ user or group profiles need to be configured to set the privilege level to 15. Console access may still require the use of an enable password. If using Cisco Secure Access Control Server (ACS), each user can be configured with a unique enable password. User profiles may also be configured to use the authentication password as enable.

The following configuration fragment illustrate the use of TACACS+:

```
! Enable AAA
aaa new-model
!
! Ensure common session ID
aaa session-id common
!
! Define server attributes
tacacs-server host <TAC+server1> single-connection key <strong-key>
tacacs-server host <TAC+server2> single-connection key <strong-key>
!
! Define server group
aaa group server tacacs+ <AAA-group>
 server <TAC+server1>
 server <TAC+server2>
!
! Define the source interface to be used to communicate with the TACACS+ servers
ip tacacs source-interface <Loopback or OOB interface>
!
! Set method list to enable login authentication
aaa authentication login <authen-exec-list> group <AAA-group> local-case
!
! Authenticate enable access
aaa authentication enable default group <AAA-group> enable
!
! Define method list to enforce exec authorization
aaa authorization exec <author-exec-list> group <AAA-group> if-authenticated
!
! Enforce console authorization
aaa authorization console
!
! Define method list to authorize the execution of administrative level commands
aaa authorization commands 15 <author-15-list> group <AAA-group> none
!
! Enable accounting
aaa accounting send stop-record authentication failure
aaa accounting exec default start-stop group <AAA-group>
aaa accounting commands 15 default start-stop group <AAA-group>
aaa accounting system default start-stop group <AAA-group>
!
! Enforce method lists to console and vty access lines
line con 0
 login authentication <authen-exec-list>
```

```
!
line vty 0 4
 authorization exec <author-exec-list>
 login authentication <authen-exec-list>
 authorization commands 15 <author-15-list>
!
```

Secure Administrative Access

Follow these best practices for securing administrative access:

- Enable SSH access when available rather the unsecure Telnet. Use at a minimum 768-bit modulus size.
- Avoid HTTP access. If possible use HTTPS instead of clear-text HTTP.
- Disable unnecessary access lines. Disabled those ports that are not going to be used with the **no exec** command.
- Per used line, explicitly define the protocols allowed for incoming and outgoing sessions. Restricting outgoing sessions prevent the system from being used as a staging host for other attacks. It should be noted, however, that outgoing Telnet may be required to manage integrated modules such as the Cisco IPS Network Module for Cisco ISR routers.
- 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 and trusted systems.
- Set idle and session timeouts—Set idle and session timeouts in every used line. Enable TCP keepalives to detect and close hung sessions.

Note HTTP access uses default login authentication and default exec authorization. In addition, privilege level for the user must be set to level 15.

Note CS-MARS SSH device discovery does not support 512-byte keys. For compatibility, use SSH modulus size equal to or larger than 768 bits.

The following configuration fragments illustrate the best practices for enabling SSH access:

```
! Prevent hung sessions in case of a loss of connection
service tcp-keepalives-in
!
! Define access class ACL to be used to restrict the sources of SSH sessions.
access-list <ACL#1> remark ACL for SSH
access-list <ACL#1> permit tcp <NOC-subnet1> <inverse-mask> any eq 22
access-list <ACL#1> permit tcp <NOC-subnet2> <inverse-mask> any eq 22
access-list <ACL#1> deny ip any any log-input
!
! ACL for last resort access
access-list <ACL#2> permit tcp host <management-station> any eq 22
access-list <ACL#2> deny ip any any log-input

! Configure a hostname and domain name
hostname <hostname>
```

```
ip domain-name <domain-name>
!
! Generate an RSA key pair, automatically enabling SSH.
crypto key generate rsa
!
! SSH negotiation timeout of 30 seconds
ip ssh timeout 30
!
! SSH authentication attempts of 2 before an interface reset
ip ssh authentication-retries 2
!
! Enforce line access class ACL, access methods and timeouts for VTYs 0 to 3.
line vty 0 3
 access-class <ACL#1> in
!
! Incoming access via SSH only
 transport input ssh
!
! 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 3 minutes
 exec-timeout 3 0
!
! Enforce access of last resource on VTY 4.
line vty 4
 access-class <ACL#2> in
 transport input ssh
 transport output none
 transport preferred none
 session-timeout 3
 exec-timeout 3 0
!
```

The following configuration fragments illustrate the best practices for enabling HTTPS access.

```
! Enforce default login authentication and exec authorization
aaa authentication login default group <AAA-group> local-case
aaa authorization exec default group <AAA-group> local
!
! Define ACL to control the sources for HTTPS sessions
access-list <ACL#> permit <NOC-subnet> <inverse-mask>
access-list <ACL#> deny   any log
!
! Disable HTTP and enable HTTPS
no ip http server
ip http secure-server
!
! Enforce HTTPS ACL and enable AAA
ip http access-class <ACL#>
ip http authentication aaa
!
! Restrict access to telnet. HTTPS access mode uses they telnet keyword.
line vty 0 4
 transport input telnet
```

For configuration guidance for Telnet and HTTP, refer to the *Network Security Baseline* document at the following URL:

http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/Baseline_Security/securebasebook.html

Routing Infrastructure Best Practices

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.

The Cisco SAFE design blueprints make use of the following measures to effectively secure the routing plane:

- *Restrict routing protocol membership*— Limit routing sessions to trusted peers, validate origin, and integrity of routing updates.
- *Control route propagation*—Enforce route filters to ensure only valid routing information is propagated. Control routing information exchange between routing peers and between redistributing processes.
- *Log status changes*—Log the status changes of adjacency or neighbor sessions.

Restrict 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, the 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:

- Enable neighbor authentication to ensure the authenticity of routing neighbor and the integrity of their routing updates. Available for BGP, IS-IS, OSPF, RIPv2 and EIGRP. Use Message Digest Algorithm Version 5 (MD5) authentication rather than insecure plain text authentication. To function properly, neighbor authentication must be enabled on both ends of the routing session.
- Use the **passive-interface default command** when enabling routing on network ranges matching a large number of interfaces. The **passive-interface default** command changes the configuration logic to a default passive, preventing the propagation of routing updates on an interface unless the interface is expressly configured with the the **no passive-interface** command. This allows to selectively enable the propagation of routing updates over the interfaces that are expected to be part of the routing process.
- When using BGP, enable TTL security check, also known as Generalized TTL Security Mechanism (GTSM, RFC 3682). 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. To work properly, TTL security check must be configured on both ends of the BGP session.

Note The effects of the **passive-interface** command 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.

Note TTL security check needs to be enabled at both ends of the peering session, otherwise BGP sessions will not be established.

The following configuration fragment shows how to enable OSPF MD5 neighbor authentication on an IOS router.

```
! 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
```

The following configuration template shows the configuration of EIGRP MD5 neighbor authentication on an IOS router. Note that EIGRP MD5 authentication is enabled on an interface or subinterface, and once configured 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.

```
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>
!
```

The following example shows the configuration of BGP MD5 neighbor authentication on an IOS router. 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 interrupts routing communications on your network.

```
router bgp <AS>
  no synchronization
  bgp log-neighbor-changes
  network <network>
  neighbor <peer-IP-address> remote-as <AS>
  neighbor <peer-IP-address> password <strong-password>
!
```

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

```
router bgp as-number
 neighbor ip-address ttl-security hops hop-count
```

Control Route Propagation

Route filtering is another important 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 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 chapter.

- *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; for example, 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.
- If route redistribution is required, enforce redistribution filters to strictly control which routes are advertised. Implementing route redistribution filters helps contain the effects of the potential injection of invalid routes, prevents loops, and helps maintain network stability.
- *Enforce route filters at stub routers*—Branches and remote locations with stub networks, enforce route filters to prevent the propagation of invalid routing information.
- *Neighbor logging*—Enable the logging of status changes of neighbor sessions on all routers.

The following example illustrates the use of inbound filters at the WAN edge:

```
! 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>
```

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>
```

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
```

Logging of Status Changes

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.

Status change message logging is disabled by default in BGP; to enable it, use the **bgp log-neighbor-changes** router command. By default, EIGRP and OSPF log status changes. If disabled, it can be enabled with use the **eigrp log-neighbor-changes** router command for EIGRP and the **log-adjacency-changes** router command for OSPF.

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

```
router bgp 10
bgp log-neighbor-changes
```

Device Resiliency and Survivability Best Practices

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 the following 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:

- Disable unnecessary services
- Infrastructure protection ACLs
- Control plane policing (CoPP)
- Port security
- Redundancy

Disable 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 therefore 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.

Note As an alternative, the Cisco IOS software provides the **AutoSecure** CLI command that helps disable these unnecessary services, while enabling other security services.

Note Before disabling a service, ensure the service is not needed.

The following are some general best practices:

- *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.
- *Global services disabled by default*—Unless explicitly needed, ensure finger, identification (identd), and TCP and UPD small servers remain disabled on all routers and switches.
- *Global services enabled by default*—Unless explicitly needed, BOOTP, IP source routing, and PAD services should be disabled globally on all routers.
- *IP directed broadcast*—Ensure directed broadcasts remain disabled on all interfaces.
- *When to disable CDP*—Disable CDP on interfaces where the service may represent a risk; for example, on external interfaces such as those at the Internet edge, and data-only ports at the campus and branch access.
- *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.

The following is an example of the show **control-plane host open-port**s command:

```
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 the 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, refer to Chapter 5, "Network Telemetry," of the *Network Security Baseline* document at the following URL: https://www.cisco.com/en/US/docs/solutions/Enterprise/Security/Baseline_Security/sec_chap5.html#wp1057909.

```
! Global Services disabled by default
 no ip finger
 no ip identd
 no service tcp-small-servers
 no service udp-small-servers
!
! Disable BOOTP, IP Source Routing and PAD global services
 no ip source-route
 no ip bootp server
 no service pad

! Disable IP  directed broadcasts on all interfaces
 interface <interface-type/number>
 no ip directed-broadcast
```

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

! Disable MOP, IP Redirects,
 interface <interface-type/number>
 no mop enabled
 no ip redirects
```

```
no ip proxy-arp
```

Infrastructure Protection ACLs (iACLs)

Infrastructure protection access control lists (iACLs) is an access control technique that shields the network infrastructure from internal and external attacks. The iACLs 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.

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. In an enterprise, iACLs may be deployed at the many network edges:

- *WAN edge*—Protecting the core infrastructure from possible threats coming from remote branch offices and partner locations.
- *Campus/Branch access*—Protecting the infrastructure from possible attacks originated from the LANs.
- *Internet edge*— Edge filters may be designed to function as an iACL to shield the infrastructure from external threats.

While there is a common structure for building iACLs, the actual ACL entries will vary dramatically depending the environment. 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 self-inflecting DoS condition. For this reason, the best approach to building an iACL is to start with a discovery ACL to identify the traffic patterns and not to control access. The iACL should only be enforced once the protocols and ports legitimately used by the infrastructure are well understood. It is also recommended to start with a relaxed iACL first, and then adjust the entries to make it more granular as the effects of the iACL are monitored.

Chapter 4 of the *Network Security Baseline* describes the iACL structure and recommended methodology. Chapter 8 of this document provides a practical example of building an iACL from a discovery ACL.

http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/Baseline_Security/securebasebook.html

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 uses the modular QoS command-line interface (MQC) for its policy configuration. MQC allows the separation of traffic into classes, and allows the user to 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 use 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.

Similarly to iACLs, while there is a common structure for configuring CoPP classes, the actual traffic classes and policers will vary dramatically depending the environment. Implementing CoPP without the proper understanding of the protocols and the devices involved may end up being ineffective and may even result in a self-inflecting DoS condition. For this reason the best approach to COPP is to start with a discovery ACL to identify the traffic classes. CoPP policies should only be enforced once the protocols and ports legitimately used by the infrastructure are well understood. It is also recommended to start by not enforcing any rate limits to each one of the traffic classes, and to configure them gradually as the effects of CoPP are monitored.

Chapter 4 of the *Network Security Baseline* describes the CoPP structure and recommended methodology.

http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/Baseline_Security/securebasebook.html

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 serverfarm, a lobby, or a Demilitarized Network (DMZ).

Typical deployment scenarios consist of the following:

- 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 serverfarm 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.
- A Voice-over-IP (VoIP) deployment, where a port may have Port Security enabled with the maximum number of MAC addresses defined as three. One MAC address is required for the workstation, and depending on the switch hardware and software one or two MAC addresses may be required 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.

In Cisco IOS, Port Security can be enabled on an interface using the **switchport port-security** command. 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.

```
interface gigabitethernet0/1
 switchport port-security maximum 3
 switchport port-security violation restrict
 switchport port-security
 switchport port-security aging time 2
 switchport port-security aging type inactivity
```

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 lobby environment.

```
interface gigabitethernet0/2
 switchport port-security maximum 1
 switchport port-security mac-address 1000.2000.3000
 switchport port-security violation restrict
 switchport port-security
```

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 SAFE design blue prints are built with a wide range of options for redundancy:

- *Backup and redundant interfaces*
- *Element redundancy*—Use of redundant processors and modules.
- *Standby devices*—Active-standby and active-active failover, first the redundancy protocols such as HSRP, VRRP, and GLBP.
- *Topological redundancy*—Designs built with redundant paths at both network and data-link layers.

Network Telemetry Best Practices

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 given time. Network telemetry offers extensive and useful detection capabilities that 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 the following:

- 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 whitepaper 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

Time Synchronization (NTP)

Time synchronizations is critical for event analysis and correlation, thus enabling NTP on all infrastructure components is a fundamental requirement.

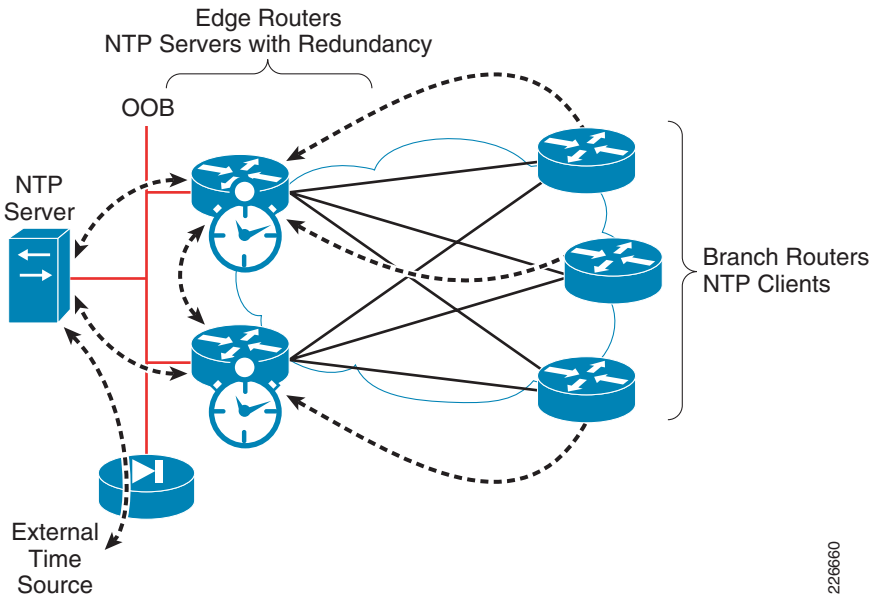
When implementing NTP, considered the following best common practices:

- Prefer a hierarchical NTP design versus a flat 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.
- 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.

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, there is likely an internal time servers at a secured segment. Unless there is 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 2-1.

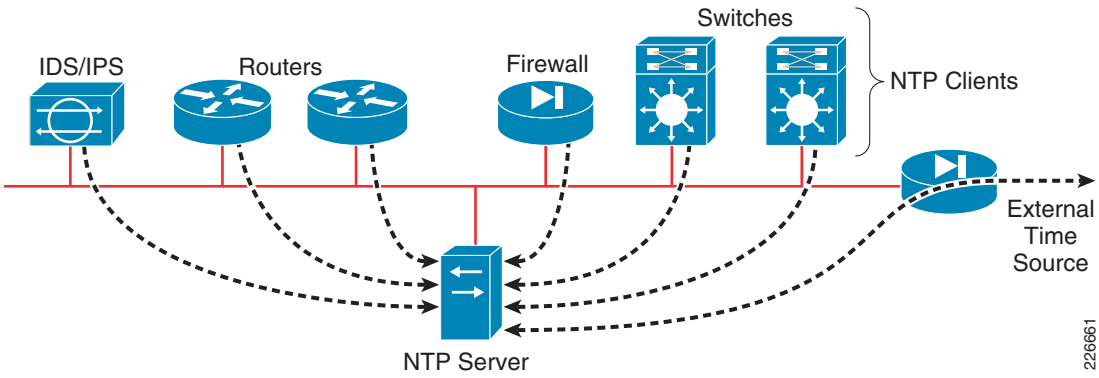
Figure 2-1 NTP Design for the WAN Edge and Remote Offices



NTP Design at the Headquarters

At the headquarters or main office, an existing OOB management network can be used. 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 2-2.

Figure 2-2 NTP Design Leveraging an OOB Management Network



The following configuration fragments illustrate the configuration of NTP client:

```
! 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 GMT
```

```
clock timezone GMT 0
!
! To periodically update the hardware clock, if present
ntp update-calendar
!
! Sets source IP address
ntp source <loopback or OOB interface>
!
! Defines servers
ntp server <NTP-Server1>
ntp server <NTP-Server2>
!
! Enables authentication
ntp authentication-key 10 md5 <strong-key>
ntp trusted-key 10
ntp authenticate
```

The following configuration fragments illustrate the configuration of NTP server:

```
! 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 GMT
clock timezone GMT 0
!
! To periodically update the hardware clock, if present
ntp update-calendar
!
! Sets source IP address
ntp source <loopback or OOB interface>
!
!Restrict the IP addresses of the servers and peers this server will communicate with.
access-list <ACL#1> remark ACL for NTP Servers and Peers
access-list <ACL#1> permit <NTPpeer1>
!
ntp access-group peer <ACL#1>
!
! Restrict the IP addresses of the clients that can communicate with this server.
access-list <ACL#2> remark ACL for NTP Client
access-list <ACL#2> permit <Client>
access-list <ACL#2> deny any log
!
ntp access-group serve-only <ACL#2>
!
! Enables authentication
ntp authentication-key 10 md5 <strong-key>
ntp trusted-key 10
ntp authenticate
!
! Defines server and peer
ntp server <NTPserver1>
ntp peer <NTPpeer1>
```

For more information on NTP design best practices, refer to *Network Time Protocol: Best Practices White Paper*

http://www.cisco.com/en/US/tech/tk869/tk769/technologies_white_paper09186a0080117070.shtml

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 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.

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.

```
interface <interface-type number>
load-interval 60
```

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>
```

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 verification
 IP verify source reachable-via ANY
  794407 verification drops
  1874428129 suppressed verification drops
```

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. This command is very useful for general troubleshooting, as well as for detecting anomalies.

```
Router#show ip traffic
```

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
```

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 **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 the 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.

Memory and CPU Threshold Notifications

Cisco IOS offers the ability to send a notification upon CPU and memory thresholds being exceeded:

- *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 percent of the total memory. Use the **show memory** command to see the total memory and available free memory.
- *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.
- *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.

The following configuration fragment illustrates the above concepts:

```
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 processor threshold is set to 91,523 Kilobytes. The total system I/O memory is 67,108,864 bytes, therefore the threshold is set to 6,710 Kilobytes:

```
memory free low-watermark processor 91523
memory free low-watermark io 6710
memory reserve critical 1000

snmp-server enable traps cpu threshold
snmp-server host <SNMP-station> traps <SNMP-community> cpu
```

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.

Step 2 Enable system message logging to a local buffer. This allows accessing 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 that newer messages overwrite older messages after the buffer is filled.

Step 3 Set the severity level of messages to be logged. Messages at or numerically lower than the specified level are logged. With respect to the severity level, the more information is logged the better; therefore, logging messages of all severity levels would be ideal. However, this may result in an overwhelming volume of messages. A 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 Set the source IP address of syslog messages to the address of an administrative loopback interface or OOB interface.

Step 5 Disable the logging of messages to the console. This helps keep the console free of messages.

```
! Enable timestamps for debugging and logging messages.
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
!
! Enable system message logging to a local buffer.
logging buffered
!
! Logging for critical equipment.
logging trap informational
logging rate-limit 1 except 3
!
! Logging for non-critical equipment.
logging trap critical
!
! Define the syslog servers to be used.
logging facility <syslogserver>
!
! Set the source IP address of syslog messages.
! logging source-interface <loopback or OOB interface>
```

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 it 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.

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.

Step 2 If using SNMPv3 (recommended), enforce an SNMP view that restricts the download of full IP routing and ARP tables.

Step 3 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.

Step 4 Set the source IP address for SNMP traps to the address used on the administrative loopback interface of OOB interface.

The following configuration example is for SNMPv1 restricting access to read-only and from a single SNMP host. For SNMPv3 configuration, refer to the *Network Security Baseline*.

```
access-list <ACL#> remark ACL for SNMP access to device
access-list <ACL#> permit <SNMP-host>
```

```
access-list <ACL#> deny   any log
snmp-server community <SNMP-Community>  RO <ACL#>
```

Network Policy Enforcement Best Practices

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 the following:

- Access edge filtering
- IP spoofing protection

Access Edge Filtering

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). For details about iACLs, refer to "Infrastructure Protection ACLs (iACLs)" section on page 2-14.

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*.

Spoofed traffic with an invalid source IP address may include traffic from either of the following:

- 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.

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. Chapter 6, "Enterprise Internet Edge" and Chapter 7, "Enterprise WAN Edge" provide the guidelines for edge ACLs designed for IP spoofing protection.

- *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*

  This feature 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

Deploying uRPF at the Internet edge as shown in the following example:

```
! Configure uRPF strict mode on the internal interfaces
 interface <Type Number>
 ip verify unicast source reachable-via rx
!
! Configure uRPF loose mode on Internet facing interfaces
 interface <Type Number>
 ip verify unicast source reachable-via any
```

Switching Infrastructure Best Practices

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 the following:

- Restrict broadcast domains
- Spanning Tree Protocol (STP) security
- Port Security
- 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. A hierarchical design like the one proposed in the campus design 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.

Spanning Tree Protocol Security

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.

STP is a useful protocol but, unfortunately, the existing 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 developed to address its risks. Not using STP would result in a loop becoming another attack vector.

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
- Enable traffic storm control

```
! Disable dynamic trunking on all switching access lines
 interface type slot/port
 switchport mode access
!
```

```
! Enable BPDU guard on end user ports and other ports not expected to participate in
Spanning Tree
 interface type slot/port
 spanning-tree portfast
 spanning-tree bpduguard enable
!
! 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
 interface type slot/port
 shutdown
 switchport access vlan <vlan_ID>
```

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. Port Security is covered in more detailed in the "Device Resiliency and Survivability Best Practices" section on page 2-12.

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
- Set the default port status to **disable**

Threats Mitigated in the Infrastructure

Table 2-1 Infrastructure Threat Mitigation Features

	Botnets	DoS on Infrastructure	DDoS on Infrastructure	Unauthorized Access	Intrusions	Routing Protocol Attacks	L2 Attacks	Visibility	Control
AAA				Yes	Yes			Yes	Yes
SNMP Authentication				Yes	Yes			Yes	Yes
SSH				Yes	Yes			Yes	Yes
Strong Password Policy				Yes	Yes				Yes
Session ACLs		Yes	Yes	Yes	Yes			Yes	Yes
Router Neighbor Authentication		Yes		Yes		Yes			Yes
Route Filtering		Yes		Yes		Yes			Yes
iACL	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes
CoPP	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes
System and Topological Redundancy	Yes	Yes	Yes			Yes	Yes		Yes