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.

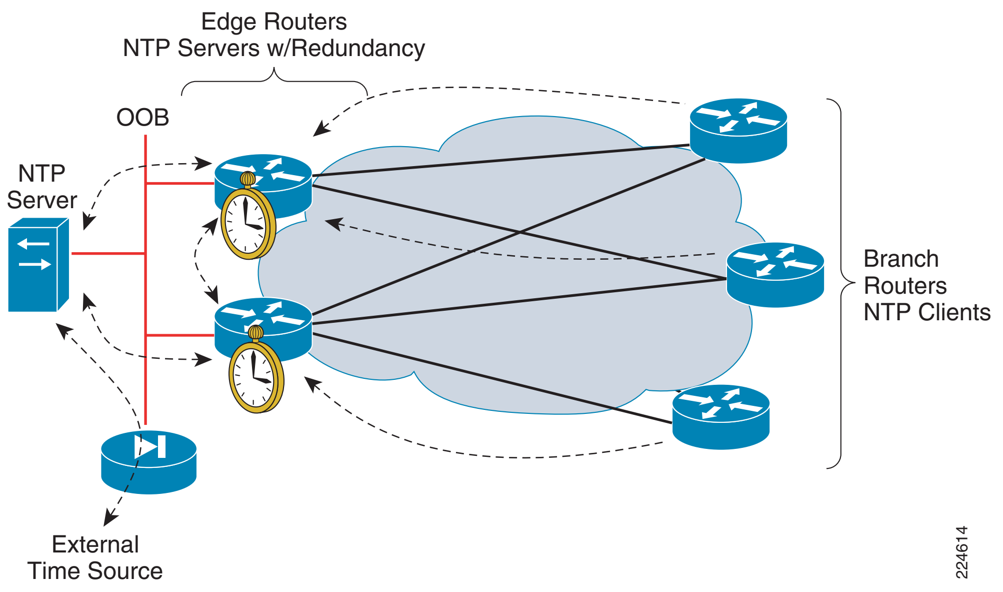
When implementing NTP follow these principles:

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.

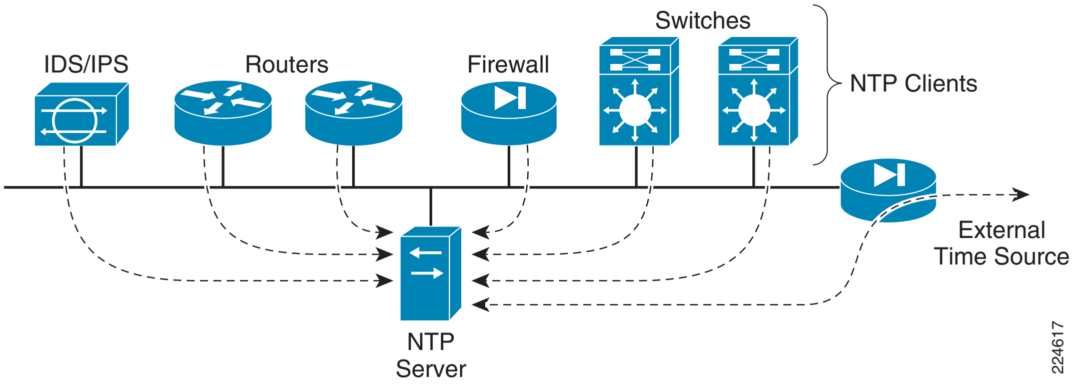
Figure 8-1 NTP Design for the WAN Edge and Remote Offices



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>
```