Routers and Routing Protocol Hardening

CCNP ROUTE: Implementing IP Routing

Cisco | Networking Academy®
Mind Wide Open™

Chapter 8 Objectives

This chapter covers the following topics:

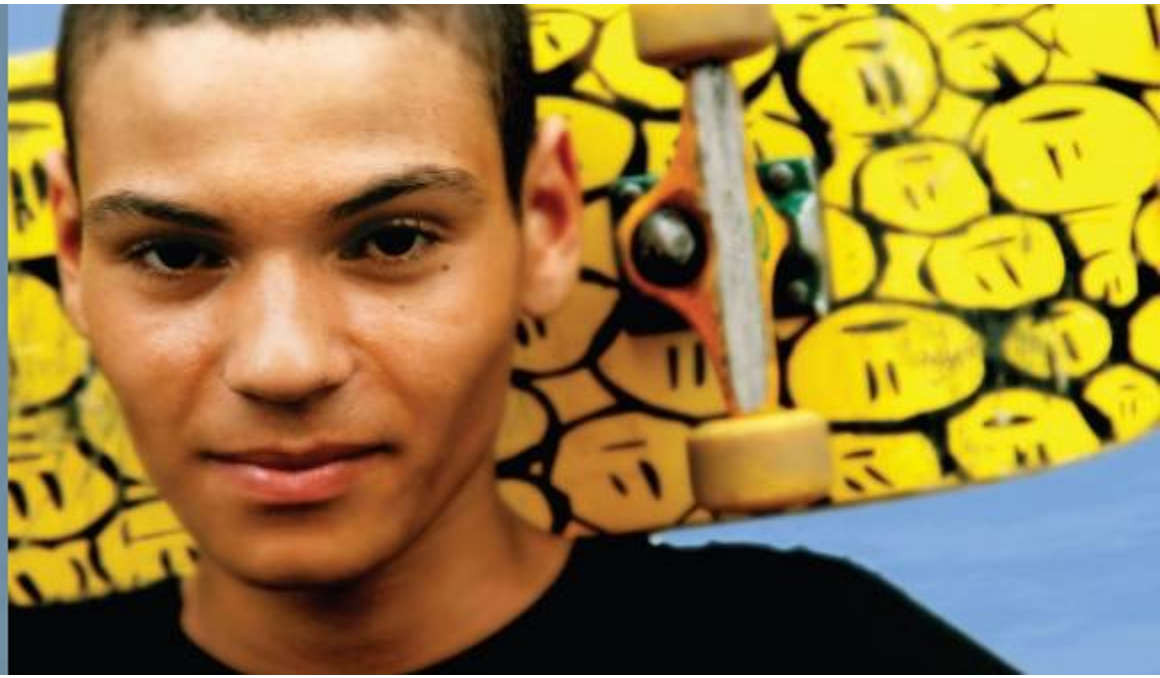
- Securing the Management Plane on Cisco Routers
- Describing Routing Protocol Authentication
- Configuring Authentication for EIGRP
- Configuring Authentication for OSPFv2 and OSPFv3
- Configuring Authentication for BGP peers
- Configuring VRF-lite

Chapter 8 Objectives

A router's operational architecture can be categorized into three planes:

- **Management plane**
  - This plane is concerned with traffic that is sent to the Cisco IOS device and is used for device management. Securing this plane involves using strong passwords, user authentication, implementing role-based command-line interface (CLI), using Secure Shell (SSH), enable logging, using Network Time Protocol (NTP), securing Simple Network Management Protocol (SNMP), and securing system files.
- **Control plane**
  - This plane is concerned with packet forwarding decisions such as routing protocol operations. Securing this plane involves using routing protocol authentication.
- **Data plane**
  - This plane is also known as the forwarding plane because it is concerned with the forwarding of data through a router. Securing this plane usually involves using access control lists (ACLs).

Securing the Management Plane on Cisco Routers



Securing the Management Plane on Cisco Routers

Device hardening tasks related to securing the management plane of a Cisco router, including the following:

- Following the router security policies
- Securing management access
- Using SSH and ACLs to restrict access to a Cisco router
- Implement logging
- Securing SNMP
- Backup configurations
- Using network monitoring
- Disabling unneeded services

Securing the Management Plane

Step 1.

- **Follow the written router security policy.**
- The policy should specify who is allowed to log in to a router and how, who is allowed to configure and update the router, or who is allowed to perform logging and monitoring actions.
- The policy should also specify the requirements for passwords that are used to access the router.

Securing the Management Plane

Step 2.

- **Secure physical access**.
- Place the router and physical devices that connect to it in a secure locked room that is accessible only to authorized personnel.
- The room should also be free of electrostatic or magnetic interference, have fire suppression, and controls for temperature and humidity.
- Install an uninterruptible power supply (UPS) and keep spare components available.
- This reduces the possibility of a network outage from power loss.

Securing the Management Plane

Step 3.

- **Use strong encrypted passwords**
- Use a complex password with a minimum of eight characters.
- Enforce a minimum length using the security password min-length global configuration command.
- Strong passwords should generally be maintained and controlled by a centralized authentication, authorization, and accounting (AAA) server.
- Some local passwords and secret information may be required, for local fallback in case AAA servers become unavailable, such as special-use usernames, secret keys, and other password information.
- Such local passwords should be properly encrypted to secure them from prying eyes.

Securing the Management Plane

Step 4.

- **Control the access to a router.**
- Console and auxiliary ports: These ports are used to gain access when a physical connection to the router is available in the form of a terminal.
- vty lines: Access to a router using SSH or Telnet is by far the most common administrative tool. For this reason, vty access should be protected using only SSH from authorized IP addresses identified in an ACL.

Securing the Management Plane

Step 5.

- **Secure management access**
- Only authorized individuals should have access to infrastructure devices.
- For this reason, configure authentication, authorization, and accounting (AAA) to control who is permitted to access a network (authenticate), what they can do on that network (authorize), and audit what they did while accessing the network (accounting).
- Authentication can be performed locally or by using a AAA authentication server.

Securing the Management Plane

Step 6.

- **Use secure management protocols.**
- Always use secure management protocols including SSH, HTTPS, and SNMPv3.
- If unsecure management protocols such as Telnet, HTTP, or SNMP must be used, then protect the traffic using an IPsec virtual private network (VPN).
- Also protect management access to the router by configuring ACLs that specify authorized hosts that can access the router.

Securing the Management Plane

Step 7.

- **Implement system logging**
- System logging provides traffic telemetry, which helps detect unusual network activity and network device failures.
- Traffic telemetry is implemented by using various mechanisms such as syslog logging, SNMP traps, and NetFlow exports.
- Use the **service timestamps log datetime** global configuration command to include date and time in the log messages.
- When implementing network telemetry, it is important that the date and time is both accurate and synchronized across all network infrastructure devices.
- This is achieved using Network Time Protocol (NTP). Without time synchronization, it is very difficult to correlate different sources of telemetry.

Securing the Management Plane

Step 8.

- **Periodically back up configurations**
- A backed-up configuration allows a disrupted network to recover very quickly.
- This can be achieved by copying a configuration to an FTP (or TFTP) server at regular intervals or whenever a configuration change is made.

Securing the Management Plane

Step 9.

- **Disable unneeded services**
- Routers support many services.
- Some of these services are enabled for historical reasons, but are no longer required today.
- Services that are not needed on the router can be used as back doors to gain access to it and should therefore be disabled.

Router Security Policy

The router security policy should help answer the following questions regarding:

- **Password encryption and complexity settings**
- **Authentication settings**
- **Management access settings**
- **Securing management access using SSH**
- **Unneeded services settings**
- **Ingress/egress filtering settings**
- **Routing protocol security settings**
- **Configuration maintenance**
- **Change management**
- **Router redundancy**
- **Monitoring and incident handling**
- **Security updates**

Use Strong Passwords

- Use a password length of ten or more characters. A longer password is a better password.
- Make passwords complex. Include a mix of uppercase and lowercase letters, numbers, symbols, and spaces.
- Avoid passwords based on repetition, dictionary words, letter or number sequences, usernames, relative or pet names, biographical information, such as birthdates, ID numbers, ancestor names, or other easily identifiable pieces of information.
- Deliberately misspell a password (for example, Smith = Smyth = 5mYth or Security = 5ecur1ty).
- Change passwords often. If a password is unknowingly compromised, the window of opportunity for the attacker to use the password is limited.
- Do not write passwords down and leave them in obvious places, such as on the desk or monitor.

Encrypting Passwords

- Encrypting Privileged EXEC Password
  - **`enable secret`** *password* global configuration command. IOS 15.0(1)S and later default to the SHA256 hashing algorithm.
  - Earlier IOS versions use the weaker message digest 5 (MD5) hashing algorithm.
- Encrypting Console and vty Passwords
  - When defining a console or vty line password using the **`password line`** command, the passwords are stored in clear text in the configuration.
  - To create local database entry encrypted to level 4 (SHA256), use the **`username`** *name* **`secret`** *password* global configuration command.
  - The **`login local`** command makes the line authenticate using the credentials configured in the local database.

Authentication, Authorization, Accounting

Implementation of the AAA model provides the following advantages:

- **Increased flexibility and control of access configuration**
- **Scalability**
- **Multiple backup systems**
- **Standardized authentication methods**

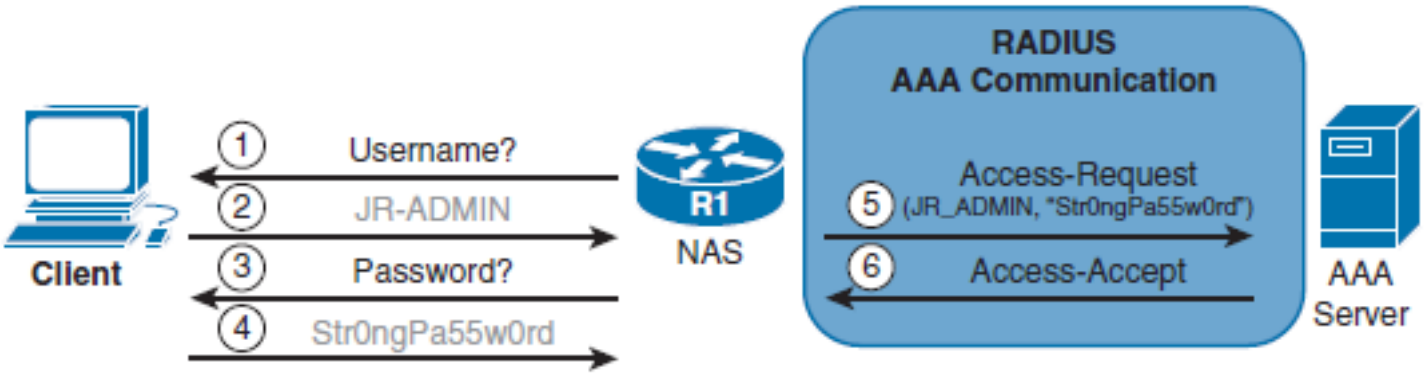
Users must authenticate against an authentication database, which can be stored:

- **Locally:** created using the **`username secret`** command
- **Centrally:** A client/server model where users are authenticated against AAA servers.

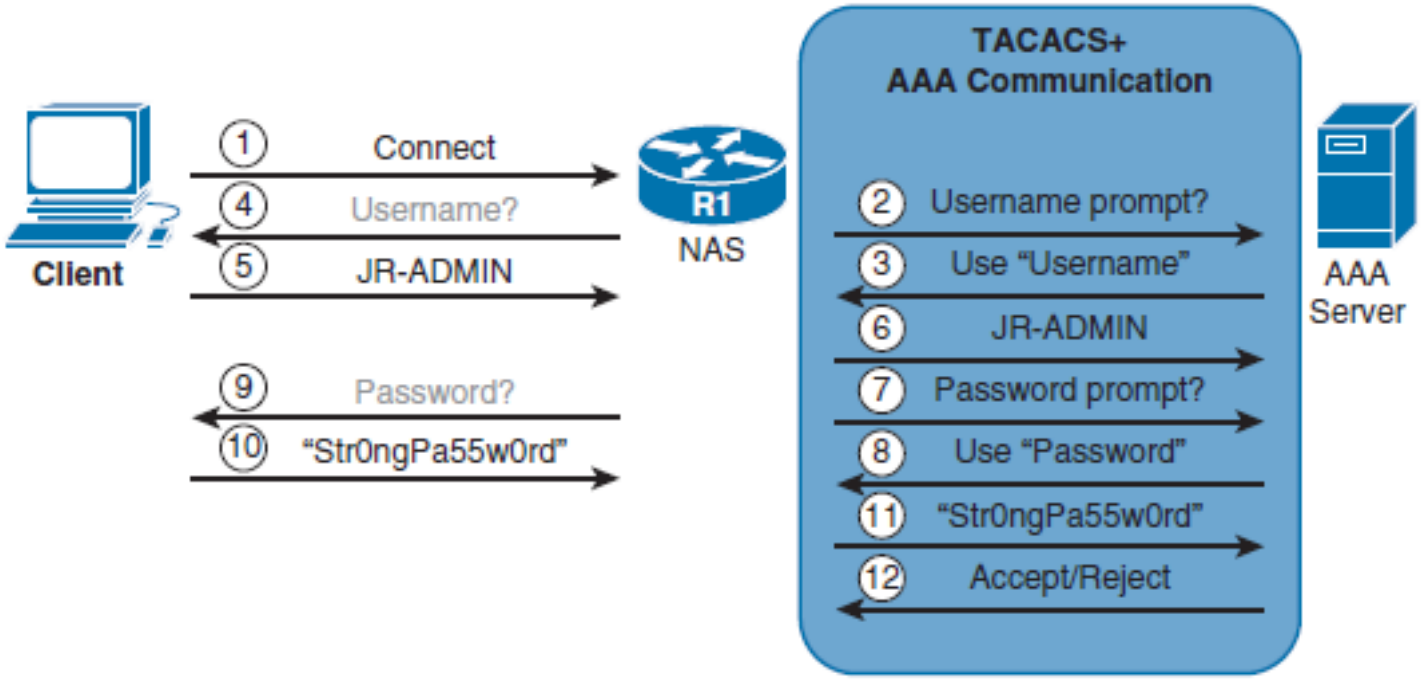
RADIUS and TACACS+ Overview

- **RADIUS protocol**
  - An open standard protocol. It combines authentication and authorization into one service using UDP port 1812 (or UDP 1645), and the accounting service uses UDP port 1813 (or UDP 1646). RADIUS does not encrypt the entire message exchanged between device and server. Only the password portion of the RADIUS packet header is encrypted.
- **TACACS+**
  - A Cisco proprietary protocol that separates all three AAA services using the more reliable TCP port 49. TACACS+ encrypts the entire message exchanged therefore communication between the device and the TACACS+ server is completely secure.

RADIUS Message Exchange



TACACS+ Message Exchange



Enabling AAA and Local Authentication

The following are the configuration steps required to enable AAA local authentication:

- **Step 1.** Create local user accounts using the **username** *name* **secret** *password* global configuration command.
- **Step 2.** Enable AAA by using the **aaa new-model** global configuration command.
- **Step 3.** Configure the security protocol parameters including the server IP address and secret key
- **Step 4.** Define the authentication method lists using the **aaa authentication login {default |** *list-name* } *method1* [...[*method4*]].
- **Step 5.** If required, apply the method lists to the console, vty, or aux lines.
- **Step 6.** (Optional) Configure authorization using the **aaa authorization** global configuration command.
- **Step 7.** (Optional) Configure accounting using the **aaa accounting** global configuration command.

Configure RADIUS Authentication with Local User for Fallback

```
R1(config)# username JR-ADMIN secret Str0ngPa55w0rd
R1(config)# username ADMIN secret Str0ng5rPa55w0rd
R1(config)#
R1(config)# aaa new-model
R1(config)#
R1(config)# radius server RADIUS-1
R1(config-radius-server)# address ipv4 192.168.1.101
R1(config-radius-server)# key RADIUS-1-pa55w0rd
R1(config-radius-server)# exit
R1(config)#
R1(config)# radius server RADIUS-2
R1(config-radius-server)# address ipv4 192.168.1.102
R1(config-radius-server)# key RADIUS-2-pa55w0rd
R1(config-radius-server)# exit
R1(config)#
R1(config)# aaa group server radius RADIUS-GROUP
R1(config-sg-radius)# server name RADIUS-1
R1(config-sg-radius)# server name RADIUS-2
R1(config-sg-radius)# exit
R1(config)#
R1(config)# aaa authentication login default group RADIUS-GROUP local
R1(config)# aaa authentication login TELNET-LOGIN group RADIUS-GROUP local-case
R1(config)# line vty 0 4
R1(config-line)# login authentication TELNET-LOGIN
R1(config-line)# exit
R1(config)#
```

Configure TACACS+ Authentication with Local User for Fallback

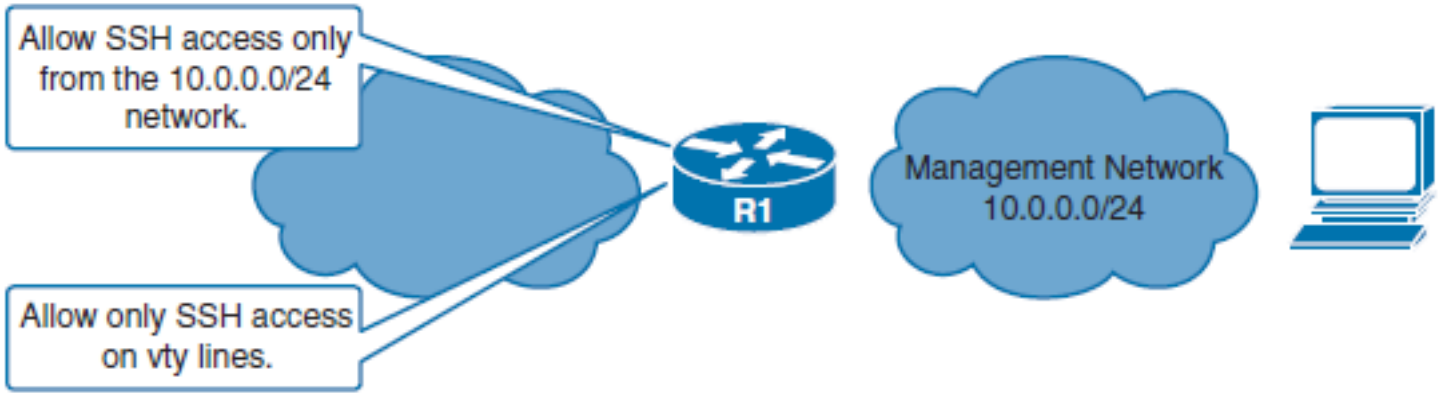
```
R1(config)# username JR-ADMIN secret Str0ngPa55w0rd
R1(config)# username ADMIN secret Str0ng5rPa55w0rd
R1(config)#
R1(config)# aaa new-model
R1(config)#
R1(config)# tacacs server TACACS-1
R1(config-server-tacacs)# address ipv4 192.168.1.201
R1(config-server-tacacs)# key TACACS-1-pa55w0rd
R1(config-server-tacacs)# exit
R1(config)#
R1(config)# tacacs server TACACS-2
R1(config-server-tacacs)# address ipv4 192.168.1.202
R1(config-server-tacacs)# key TACACS-2-pa55w0rd
R1(config-server-tacacs)# exit
R1(config)#
R1(config)# aaa group server tacacs TACACS-GROUP
R1(config-sg-tacacs+)# server name TACACS-1
R1(config-sg-tacacs+)# server name TACACS-2
R1(config-sg-tacacs+)# exit
R1(config)#
R1(config)# aaa authentication login default group TACACS-GROUP local
R1(config)# aaa authentication login TELNET-LOGIN group TACACS-GROUP local-case
R1(config)# line vty 0 4
R1(config-line)# login authentication TELNET-LOGIN
R1(config-line)# exit
R1(config)#
```

Use SSH Instead of Telnet

Complete the following steps to enable the SSH access instead of Telnet:

- **Step 1. Enable the use of SSH protocol:** Ensure that the target routers are running a Cisco IOS release that supports SSH.
- **Step 2. Enable local authentication for SSH access:** This is because SSH access requires login using username and password.
- **Step 3. Enable the use of SSH protocol:** Optionally allow SSH access only from authorized hosts by specifying an ACL.

Use SSH Instead of Telnet




```
Router(config)# hostname R1
R1(config)# ip domain-name cisco.com
R1(config)# username ADMIN privilege 15 secret class12345
```

```
R1(config)# crypto key generate rsa modulus 2048
The name for the keys will be: R1.cisco.com
% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 8 seconds)

R1(config)#
*Aug 13 17:22:58.625: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

Use SSH Instead of Telnet

```
R1(config)# ip ssh version 2
```

```
R1(config)# ip access-list standard PERMIT-SSH
R1(config-std-nacl)# remark ACL permitting SSH to hosts on the Management LAN
R1(config-std-nacl)# permit 10.0.0.0 0.0.0.255
R1(config-std-nacl)# deny any log
R1(config-std-nacl)# exit
```

```
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# access-class PERMIT-SSH in
R1(config-line)# end
R1#
```

Securing Access to the Infrastructure Using Router ACLs

- All the traffic to the IP addresses of the network infrastructure devices is dropped and logged.
  - This rule prevents the network users from sending the routing protocol or the management traffic to network devices.
  - Include the destination addresses that encompass all the device IP addresses as a condition.
- All the other traffic is permitted and allows all the transit traffic over the network.

Securing Access to the Infrastructure Using Router ACLs

```
R1(config)# ip access-list extended ACL-INFRASTRUCTURE-IN
R1(config-ext-nacl)# remark Deny IP fragments
R1(config-ext-nacl)# deny tcp any any fragments
R1(config-ext-nacl)# deny udp any any fragments
R1(config-ext-nacl)# deny icmp any any fragments
R1(config-ext-nacl)# deny ip any any fragments
R1(config-ext-nacl)# remark permit required connections for management traffic
R1(config-ext-nacl)# permit tcp host 10.10.12.2 host 10.10.12.1 eq 179
R1(config-ext-nacl)# permit tcp host 10.10.12.2 eq 179 host 10.10.12.1
R1(config-ext-nacl)# permit tcp host 10.0.0.10 any eq 22
R1(config-ext-nacl)# remark Permit ICMP Echo from management station
R1(config-ext-nacl)# permit icmp host 10.0.0.10 any echo
R1(config-ext-nacl)# remark Deny all other IP traffic to any network device
R1(config-ext-nacl)# deny ip any 10.0.0.0 0.0.0.255
R1(config-ext-nacl)# remark permit transit traffic
R1(config-ext-nacl)# permit ip any any
R1(config-ext-nacl)# exit
R1(config)# interface ethernet 0/0
R1(config-if)# ip access-group ACL-INFRASTRUCTURE-IN in
R1(config-if)#^Z
R1#
*Aug 13 18:19:57.308: %SYS-5-CONFIG_I: Configured from console by console
```

Implement Unicast Reverse Path Forwarding

- Unicast Reverse Path Forwarding (uRPF) helps limit the malicious traffic on an enterprise network.
- This security feature works with Cisco Express Forwarding (CEF) by enabling the router to verify that the source of any IP packets received is in the CEF table and reachable via the routing table. If the source IP address is not valid, the packet is discarded.
- Prevents common spoofing attacks and follows RFC 2827 for ingress filtering to defeat denial-of-service (DoS) attacks, which employ IP source address spoofing.
- RFC 2827 recommends that service providers filter their customers' traffic and drop any traffic entering their networks that is coming from an illegitimate source address.

Implement Unicast Reverse Path Forwarding

The uRPF feature works in one of two modes:

Strict mode

- The packet must be received on the interface that the router would use to forward the return packet.
- uRPF configured in strict mode may drop legitimate traffic that is received on an interface that was not the router's choice for sending return traffic.
- Dropping this legitimate traffic could occur when asymmetric routing paths are present in the network.
- Use the **`ip verify unicast source reachable-via rx`** command.

Loose mode

- The source address must appear in the routing table.
- Administrators can change this behavior using the **allow-default** option, which allows the use of the default route in the source verification process.
- In addition, a packet that contains a source address for which the return route points to the Null 0 interface will be dropped.
- An access list may also be specified that permits or denies certain source addresses in uRPF loose mode.
- Use the **`ip verify unicast source reachable-via any`** command

Enabling uRPF

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ip verify unicast source reachable-via any
R1(config-if)# exit
R1(config)#
R1(config)# interface GigabitEthernet 0/1
R1(config-if)# ip verify unicast source reachable-via rx
R1(config-if)# exit
R1(config)#
```

Implement Logging

- Network administrators need to implement logging to get insight into what is happening in their network.
- Although logging can be implemented locally on a router, this method is not scalable.
- Therefore, it is important to implement logging to external destination.

Implement Logging

- Network Time Protocol (NTP) can be used to synchronize network devices to the correct time.
- It is also important that syslog entries be stamped with the correct time and date.
- Time stamps are configured using the **service timestamps** [**debug** | **log**] [**uptime** | **datetime** [*msec*]] **[localtime**] [**show-timezone**] [**year**] global configuration command.

Implementing Network Time Protocol

- An NTP network usually gets its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server.
- NTP then distributes this time across the network using UDP port 123.

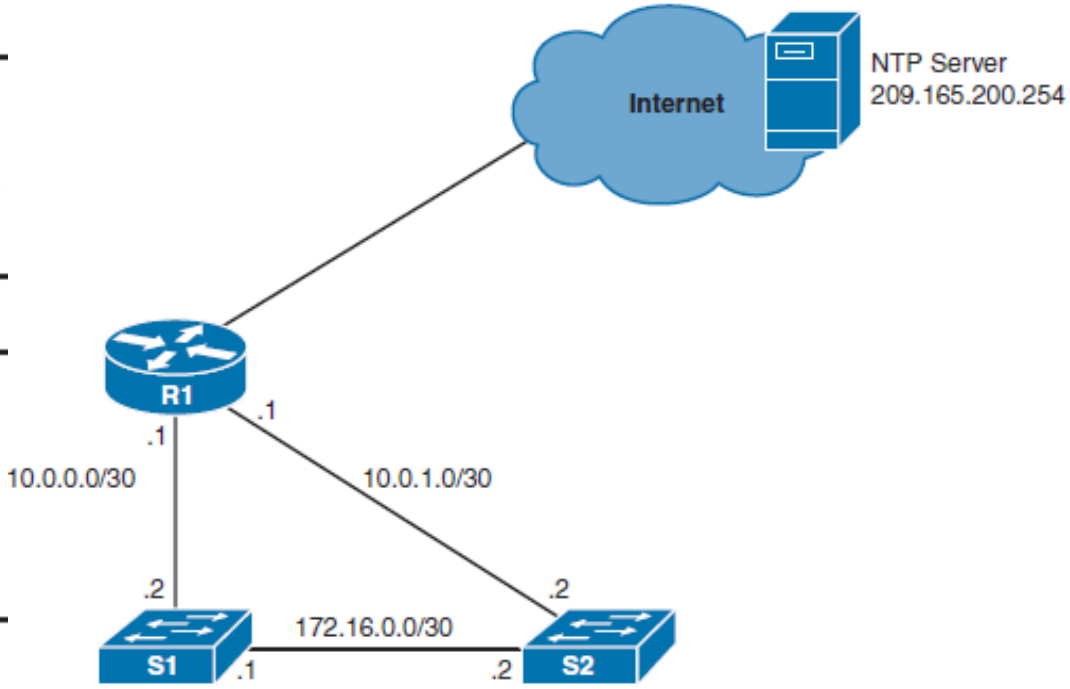
NTP Modes

- **Server:** Also called the NTP master because it provides accurate time information to clients. Configured with the **`ntp master`** [*`stratum`*] global configuration command.
- **Client:** Synchronizes its time with the NTP server. An NTP client is enabled with the **`ntp server`** { *`ntp-master-hostname`* | *`ntp-master-ip-address`* } command.
- **Peers:** Also called symmetric mode, peers exchange time synchronization information. Peers are configured using the **`ntp peer`** { *`ntp-peerhostname`* | *`ntp-peer-ip-address`* } command.
- **Broadcast/multicast:** Special "push" mode of NTP server that provides one-way time announcements to receptive NTP clients. Typically used when time accuracy is not a big concern. Configured with the **`ntp broadcast client`** interface configuration command.

Enabling NTP

```
R1(config)# ntp server 209.165.200.254
R1(config)# clock timezone EST -5
R1(config)# clock summer-time EST recurring
R1(config)#
```

```
S1(config)# ntp server 10.0.0.1
S1(config)# clock timezone EDT -5
S1(config)# clock summer-time EDT recurring
S1(config)# ntp peer 172.16.0.2
S1(config)#
```




```
S2(config)# ntp server 10.0.1.1
S2(config)# clock timezone EST -5
S2(config)# clock summer-time EST recurring
S2(config)# ntp peer 172.16.0.1
S2(config)#
```

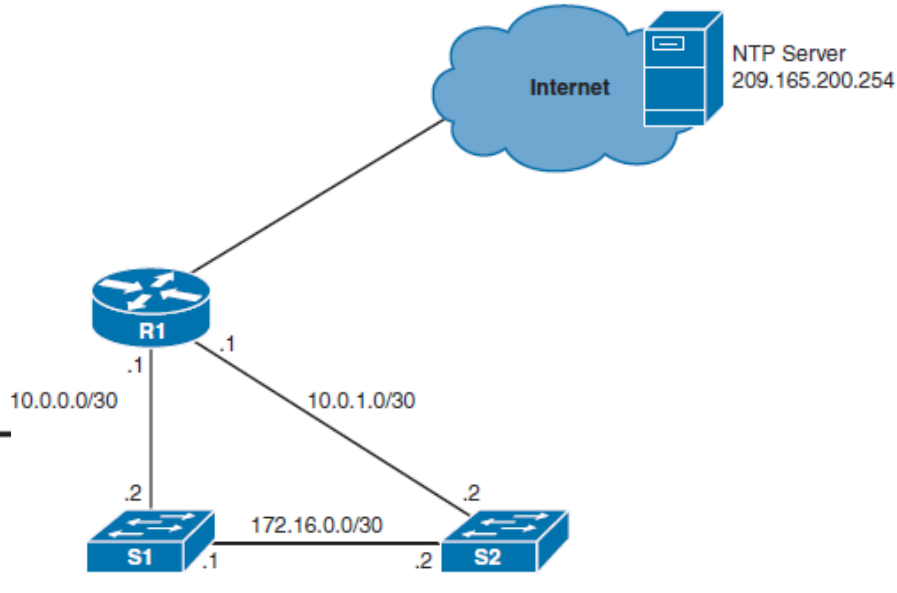
Securing NTP

- **Authentication**
  - NTP authenticates the source of the information, so it only benefits the NTP client. Cisco devices support only MD5 authentication for NTP.
- **Access control lists**
  - Configure access lists on devices that provide time synchronization to others. ACLs are applied to NTP using the **ntp access-group** { **peer** | **query-only** | **serve** | **serve-only** }

NTP Authentication Configuration

- **Step 1**
  - Define NTP authentication key or keys with the **ntp authentication-key** *key_number* **md5** *pass* global configuration command. Every number specifies a unique NTP key.
- **Step 2**
  - Enable NTP authentication using the **ntp authenticate** global configuration command.
- **Step 3**
  - Tell the device which keys are valid for NTP authentication using the **ntp trusted-key** *key* global configuration command. The *key* argument should be the key defined in Step 1.
- **Step 4**
  - Specify the NTP server that requires authentication using the **ntp server** *ip_address* **key** *key_number* global configuration command. The command can also be used to secure NTP peers.

NTP Authentication




```
R1(config)# ntp authentication-key md5 NTP-pa55w0rd
R1(config)# ntp authenticate
R1(config)# ntp trusted-key 1
R1(config)#
R1(config)# access 10 permit 10.0.0.0 0.0.255.255
R1(config)# ntp access-group serve-only 10
R1(config)#
```

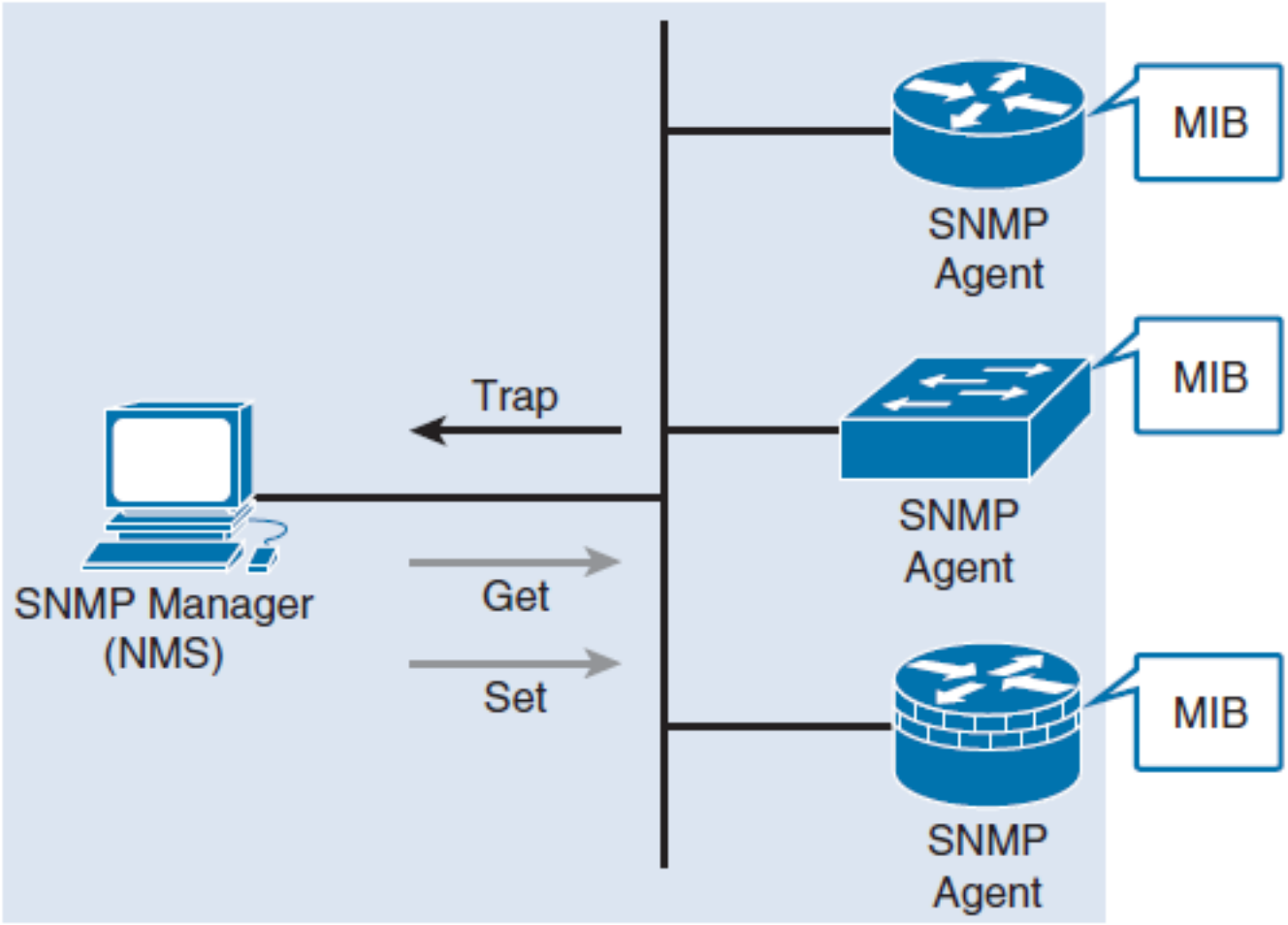
```
S1(config)# ntp authentication-key md5 NTP-pa55w0rd
S1(config)# ntp authenticate
S1(config)# ntp trusted-key 1
S1(config)# ntp server 10.0.0.1 key 1
S1(config)#
```

NTP Versions

Currently NTP Versions 3 and 4 are used in production networks. NTPv4 is an extension of NTP Version 3 and provides the following capabilities:

- Supports both IPv4 and IPv6 and is backward-compatible with NTPv3. NTPv3 does not support IPv6.
- Uses IPv6 multicast messages instead of IPv4 broadcast messages to send and receive clock updates.
- Improved security over NTPv3 as NTPv4 provides a whole security framework based on public key cryptography and standard X509 certificates.
- Improved time synchronization and efficiency.
- NTPv4 access group functionality accepts IPv6 named access lists as well as IPv4 numbered access lists.

Implementing SNMP



Implementing SNMP

SNMP defines management information between these three elements:

- **SNMP manager**
  - The SNMP manager collects information from an SNMP agent using the Get action and can change configurations on an agent using the Set action.
- **SNMP agents (managed node)**
  - Resides on the SNMP-managed networking client and responds to the SNMP manager's Set and Get requests to the local MIB.
  - SNMP agents can be configured to forward real-time information directly to an SNMP´manager using traps (or notifications).
- **Management Information Base (MIB)**
  - Resides on the SNMP-managed networking client and stores data about the device operation including resources and activity. The MIB data is available to authenticated SNMP managers.

SNMP Versions

- **SNMPv1**
  - Original version, which uses community strings for authentication. These community strings are exchanged in clear text and therefore very unsecure. SNMPv1 is considered to be obsolete.
- **SNMPv2**
  - Update to SNMPv1 that improved performance, security, confidentiality, and SNMP communications. SNMPv2c is the standard and uses the same community string authentication format of SNMPv1.
- **SNMPv3**
  - Update to SNMPv2 that adds security and remote configuration enhancements. Specifically, SNMPv3 provides authentication, message integrity, and encryption.
    - **noAuthNoPriv:** Authenticates SNMP messages using a clear-text community string
    - **authNoPriv:** Authenticates SNMP messages using either HMAC with MD5 or HMAC with SHA-1
    - **authPriv:** Authenticates SNMP messages by using either HMAC-MD5 or SHA usernames and encrypts SNMP messages using DES, 3DES, or AES

Differences Between SNMP Security Levels

SNMP Version	Security Level	Authentication	Encryption
SNMPv1	noAuthNoPriv	Community string	No
SNMPv2	noAuthNoPriv	Community string	No
SNMPv3	noAuthNoPriv	Username	No
	authNoPriv	MD5 or SHA-1	No
	authPriv	MD5 or SHA-1	DES, 3DES, or AES

SNMP Protection

- There are two types of community strings in SNMPv2:
- **Read-only (RO):** Provides access to the MIB variables, but does not allow these variables to be changed, only read. Because security is so weak in SNMPv2, many organizations only use SNMP in this read-only mode.
- **Read-write (RW):** Provides read and write access to all objects in the MIB.

If SNMPv2 is used, it should be secured by

- Using an uncommon, complex, long community string.
- Changing the community strings at regular intervals.
- Enabling read-only access only. If read write access is required, limit the read write access to the authorized SNMP manager.
- SNMP trap community names must be different than Get and Set community strings.

Sample SNMPv2 Configuration

```
R1(config)# ip access-list standard PROTECT-SNMP
R1(config-std-nacl)# remark Identify SNMP manager host
R1(config-std-nacl)# permit host 10.1.2.3
R1(config-std-nacl)# exit
R1(config)# snmp-server community R1-5ecret-5tr1ng ro PROTECT-SNMP
```

Configuring SNMPv3

- **Step 1.** Configure an ACL to limit who has access SNMP access to the device.
- **Step 2.** Configure an SNMPv3 view using the **snmp-server view** *view-name* global configuration command.
- **Step 3.** Configure an SNMPv3 group using the **snmp-server group** *group-name* global configuration command.
- **Step 4.** Configure an SNMPv3 user using the **snmp-server user** *username groupname* global configuration command.
- **Step 5.** Configure an SNMPv3 trap receiver using the **snmp-server host** global configuration command.
- **Step 6.** Configure interface index persistence using the **snmp-server ifindex persist** global configuration command.

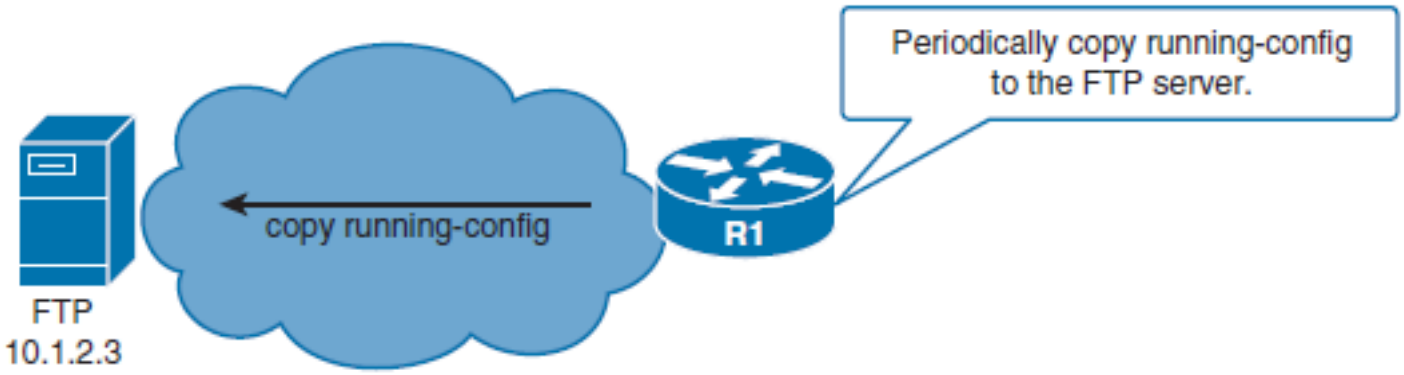
Sample SNMPv3 Configuration

```
R1(config)# ip access-list standard SNMPv3-ACL
R1(config-std-nacl)# remark ACL limits SNMP access to management network
R1(config-std-nacl)# permit 10.1.1.0 0.0.0.255
R1(config-std-nacl)# exit
R1(config)#
R1(config)# snmp-server view OPS sysUpTime included
R1(config)# snmp-server view OPS ifOperStatus included
R1(config)# snmp-server view OPS ifAdminStatus included
R1(config)# snmp-server view OPS ifDescr included
R1(config)#
R1(config)# snmp-server group MY-GROUP v3 priv read OPS write OPS access SNMPv3-ACL
R1(config)# snmp-server user ADMIN MY-GROUP v3 auth sha SNMP-Secret1 priv aes 256
SNMP-Secret2
*Nov  3 21:12:10.863: Configuring snmpv3 USM user, persisting snmpEngineBoots.
Please Wait...
R1(config)#
R1(config)# snmp-server enable traps
NHRP MIB is not enabled: Trap generation suppressed
However, configuration changes effective
R1(config)#
R1(config)# snmp-server host 10.1.1.254 traps version 3 priv ADMIN cpu
R1(config)#
R1(config)# snmp-server ifindex persist
R1(config)#
```

Verifying SNMPv3

- **`show snmp`**
  - Provides basic information about the SNMP configuration.
  - Displays SNMP traffic statistics, see whether the SNMP agent is enabled, or verify whether the device is configured to send traps, and if so, to which SNMP managers.
- **`show snmp` *`view`***
  - Provides information about configured SNMP views to verify for each group, see which OIDs are included
- **`show snmp group`**
  - Provides information about the configured SNMP groups. The most important parameters are the security model and levels.

Configuration Backups



The **archive** Command is used to perform backups automatically.

- The **path** is a required parameter that is specified by using URL notation form. It can denote either a local or a network path.
- You can use two variables with the **path** command:
  - **$h** will be replaced with device hostname.
  - **$t** will be replaced with date and time of the archive.

Archive Configuration

- Manually

```
R1(config)# archive
R1(config-archive)# path ftp://admin:cisco123@10.1.2.3/$h.cfg
R1(config-archive)# ^Z
R1#
```

```
R1# archive config
Writing R1.cfg-Sep-20-13-05-09.868-0
R1#
```

- Automatically

```
R1(config)# archive
R1(config-archive)# write-memory
R1(config-archive)# time-period 10080
R1(config-archive)# end
R1#
R1# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Writing R1.cfg-Sep-20-13-15-09.496-1
R1#
```

Verifying Archives

```
R1# show archive
The maximum archive configurations allowed is 10.
The next archive will be named ftp://admin:cisco123@10.1.2.3/R1-5
Archive #    Name
0
1            ftp://admin:cisco123@10.1.2.3/R1-1
2            ftp://admin:cisco123@10.1.2.3/R1-2
3            ftp://admin:cisco123@10.1.2.3/R1-3
4            ftp://admin:cisco123@10.1.2.3/R1-4
```

Using SCP

- The Secure Copy (SCP) feature provides a secure and authenticated method for copying router configuration or router image files.

Enabling SCP on a Router

- **Step 1.** Use the **username name** [**privilege** *level*] { **secret** *password* } command for local authentication or configure TACACS+ or RADIUS.
- **Step 2.** Enable SSH. Configure a domain name using the **ip domain-name** and generating the crypto keys using the **crypto key generate rsa general key** global configuration commands.
- **Step 3.** AAA with the **aaa new-model** global configuration mode command.
- **Step 4.** Use the **aaa authentication login** { **default** | *list-name* } *method1* [*method2* ...] command to define a named list of authentication methods.
- **Step 5.** Use the **aaa authorization** { **network** | **exec** | **commands** *level* } { **default** | *listname* } *method1...* [*method4*] command to configure command authorization.
- **Step 6.** Enable SCP server-side functionality with the **ip scp server enable** command.

Sample SCP Configuration

```
R1(config)# username ADMIN privilege 15 secret SCP-Secret
R1(config)# ip domain-name scp.cisco.com
R1(config)# crypto key generate rsa general-keys modulus 1024
The name for the keys will be: R1.scp.cisco.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 2 seconds)

R1(config)#
*Nov  3 22:25:28.135: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)# aaa new-model
R1(config)# aaa authentication login default group radius local-case
R1(config)# aaa authorization exec default group radius local
R1(config)# ip scp server enable
```

Disabling Unused Services

Service	Description of Service	Commands Used to Disable Service
DNS Name Resolution	If no DNS server is specifically mentioned in the router configuration, all the name queries are sent to the broadcast address of 255.255.255.255 by default.	Router(config)# **no ip domain-lookup**
CDP	The CDP is a proprietary protocol that Cisco devices use to identify their directly connected neighbors. CDP, like any other unnecessary local service, is considered potentially harmful to security.	Router(config)# **no cdp run** Router(config-if)# **no cdp enable**
NTP	If NTP is not used in the network, it should be disabled. You can disable the processing of NTP packets on a specific interface.	Router(config-if)# **ntp disable**
BOOTP Server	BOOTP uses UDP to formulate a network request to allow a device to obtain and configure its own IP information, such as IP address and subnet mask. However, the BOOTP protocol is seldom used, and it gives a hacker an opportunity to steal an IOS image.	Router(config)# **no ip bootp server**
DHCP	DHCP is essentially an extension of BOOTP.	Router(config)# **no ip dhcp-server**
Proxy ARP	Proxy ARP replies are sent to an ARP request destined for another device. When an intermediate Cisco device knows the MAC address of the destination device, it can act as a proxy. When an ARP request is destined for another Layer 3 network, a proxy ARP device extends a LAN perimeter by enabling transparent access between multiple LAN segments. This presents a security problem. An attacker can issue multiple ARP requests and use up the proxy ARP device's resources when it tries to respond to these requests in a DoS attack. Proxy ARP is enabled on Cisco router interfaces.	Router(config-if)# **no ip proxy-arp**
IP Source Routing	An option is found in the header of every IP packet. The Cisco IOS Software examines the option and acts accordingly. Sometimes an option indicates source routing. This means that the packet is specifying its own route. This feature poses a known security risk, such as a hacker taking control of a packet's route and directing it through the network. So, if source routing is not necessary in your network, you should disable it on all routers.	Router(config)# **no ip source-route**
IP Redirects	ICMP messages that are automatically sent by Cisco routers in response to various actions can give away a lot of information, such as routes, paths, and network conditions, to an unauthorized individual.	Router(config-if)# **no ip redirects**
HTTP Service	The Cisco IOS Software includes a web browser user interface from which you can issue Cisco IOS commands. You should disable HTTP server if it is not used.	Router(config)# **no ip http server**

Conditional Debugging

- It is practical to know how to limit debug output:
  - Use an ACL
  - Enable conditional debugging
- The **`debug ip packet`** [*`access-list`*] command displays general IP debugging and is useful for analyzing messages traveling between local and remote hosts and to narrow down the scope of debugging.
- Conditional debugging is sometimes called “conditionally triggered debugging.” It can be used to
  - Limit output based on the interface. Debugging output is turned off for all interfaces except the specified interface.
  - Enable debugging output for conditional debugging events. Messages are displayed as different interfaces meet specific conditions.
- To enable, define the condition with the **debug condition interface**

Enabling Conditional Debugging

- Commands required to debug NAT and IP packet details and limit to output for interface Fa0/0 only.

```
R1# debug condition interface fa0/0
Condition 1 set
R1# debug ip packet detail
IP packet debugging is on (detailed)
R1#
R1# debug ip nat detailed
IP NAT detailed debugging is on
R1#
```

Routing Protocol Authentication Options

Routing Protocol Authentication Options

- The purpose of routing protocol authentication
- Increasing the security of routing protocol authentication with time-based key chains
- Authentication options with different routing protocols

The Purpose of Routing Protocol Authentication

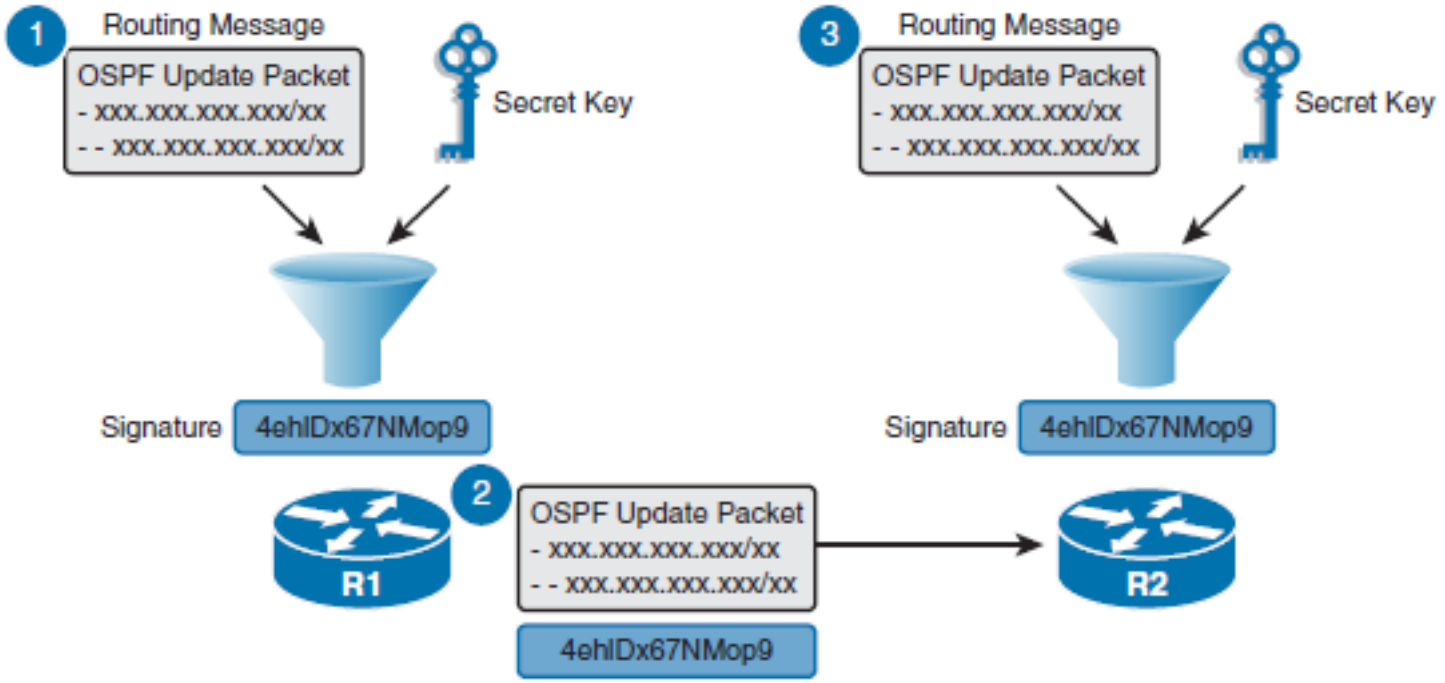
- The falsification of routing information is a more subtle class of attack that targets the information carried within the routing protocol.
- The consequences of falsifying routing information are as follows:
  - Redirect traffic to create routing loops
  - Redirect traffic to monitor on an insecure line
  - Redirect traffic to discard it
- Two types of neighbor authentication can be used:
  - Plain-text authentication
  - Hashing authentication

Plain-Text Authentication


```
R1(config)# interface ethernet 0/1
R1(config-if)# ip ospf authentication
R1(config-if)# ip ospf authentication-key PLAINTEXT
% OSPF: Warning: The password/key will be truncated to 8 characters
R1(config-if)# ip ospf authentication-key PLAINTEX
R1(config-if)#
*Sep 21 11:45:53.670: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Ethernet0/1 from
FULL to DOWN, Neighbor Down: Dead timer expired
R1(config-if)#
```

```
R2(config)# interface ethernet 0/0
R2(config-if)# ip ospf authentication
R2(config-if)# ip ospf authentication-key PLAINTEX
R2(config-if)#
*Sep 21 11:46:38.709: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Ethernet0/0 from
LOADING to FULL, Loading Done
R2(config-if)# exit
```

Hashing Authentication



Hashing Authentication

- The process can be explained in three steps:
- **Step 1.** When R1 sends a routing update to R2, it uses a hashing algorithm such as MD5 or SHA. The hashing algorithm is essentially a complex mathematical formula that uses the data in the OSPF update and a predefined secret key to generate a unique hash value (signature). The resulting signature can be derived only by using the OSPF update and the secret key that is only known to the sender and receiver.
- **Step 2.** The resulting signature is appended to the routing update and sent to R2.
- **Step 3.** When R2 receives the routing update and uses the same hashing algorithm as R1 to calculate a hash value. Specifically, it uses the data from the received OSPF update and its predefined secret key.

Time-Based Key Chains

- Key Chain Specifics:
  - **Key ID:** Configured using the **`key`** *`key-id`* `key` chain configuration mode command. Key IDs can range from 1 to 255.
  - **Key string (password):** Configured using the **`key-string`** *`password`* key chain key configuration mode command.
  - **Key lifetimes:** (Optional) Configured using the **`send-lifetime`** and **`accept-lifetime`** key chain key configuration mode commands.

Sample EIGRP Key Chain Configuration

```
R1(config)# key chain R1-Chain
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string firstkey
R1(config-keychain-key)# accept-lifetime 4:00:00 Jan 1 2015 Jan 31 2015
R1(config-keychain-key)# send-lifetime 4:00:00 Jan 1 2015 4:00:00 Jan 31 2015
R1(config-keychain-key)# exit
R1(config-keychain)# key 2
R1(config-keychain-key)# key-string secondkey
R1(config-keychain-key)# accept-lifetime 4:00:00 Jan 25 2015 Feb 28 2015
R1(config-keychain-key)# send-lifetime 4:00:00 Jan 25 2015 Feb 28 2015
R1(config-keychain-key)# end
R1#
```

Authentication Options with Different Routing Protocols

Routing Protocol	Plain Text Authentication	MD5 Hashing Authentication	SHA Hashing Authentication	Key Chain Support
RIPv2	Yes	Yes	No	Yes
EIGRP	No	Yes	Yes, using named EIGRP	Yes
OSPFv2	Yes	Yes	Yes, using key chains	Yes
OSPFv3	No	Yes	Yes	No
BGP	No	Yes	No	No

Configuring EIGRP Authentication

Configuring EIGRP Authentication

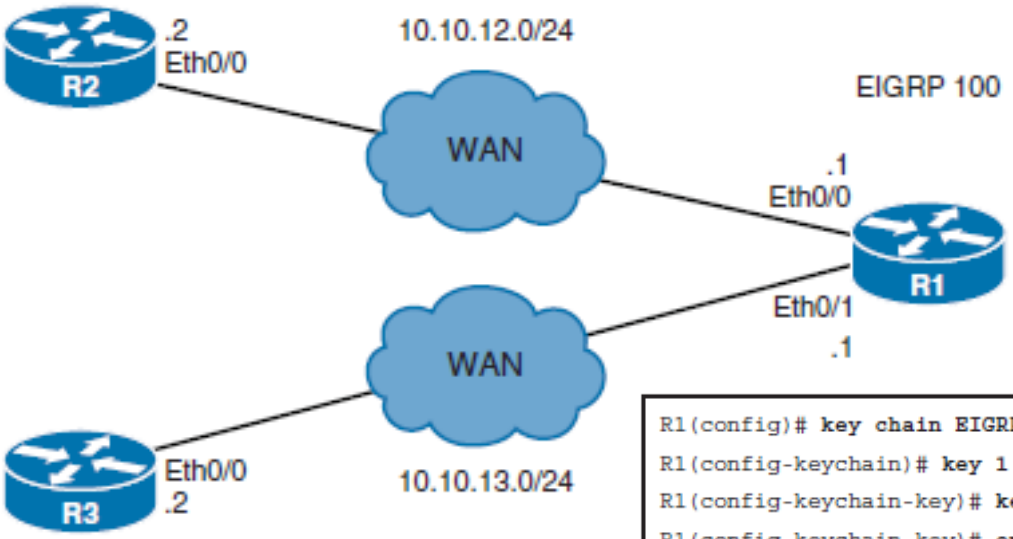
This section describes how to configure the following:

- Classic IPv4 and neighbor authentication using preshared passwords
- IPv6 EIGRP neighbor authentication using preshared passwords
- Classic IPv4 and IPv6 EIGRP neighbor authentication using the named EIGRP method

EIGRP Authentication Configuration Checklist

- **Step 1. Configure the key chain**
  - The **key chain** global configuration command is used to define all the keys that are used for EIGRP MD5 authentication.
  - Once in key chain configuration mode, use the **key** command to identify the key in the key chain.
  - When the **key** command is used, the configuration enters the key chain key configuration mode, where the **key-string** *authentication-key* configuration command must be used to specify the authentication string (or password).
  - The key ID and authentication string must be the same on all neighboring routers.
- **Step 2. Configure the authentication mode for EIGRP**
  - The only authentication type that is available in classic EIGRP configuration is MD5. The newer named EIGRP configuration method also supports the more secure SHA hashing algorithm.
- **Step 3. Enable authentication to use the key or keys in the key chain**
  - Authentication is enabled using the **ip authentication key-chain eigrp** interface command.

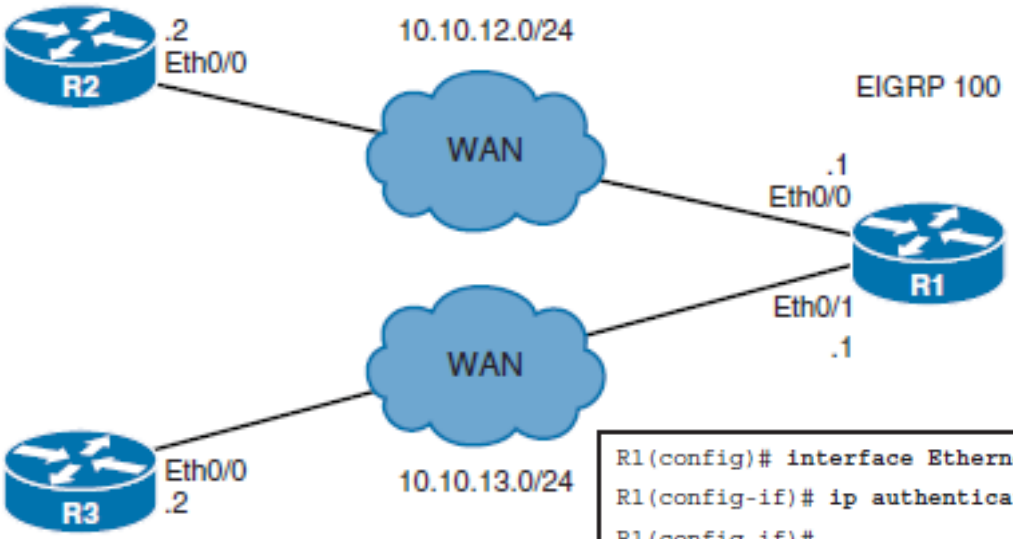
Configuring EIGRP Authentication




```
R1(config)# key chain EIGRP-KEYS
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string secret-1
R1(config-keychain-key)# end
R1# show key chain
Key-chain EIGRP-KEYS:
    key 1 -- text "secret-1"
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]
R1#
```

```
R2(config)# key chain EIGRP-KEYS
R2(config-keychain)# key 1
R2(config-keychain-key)# key-string secret-1
R2(config-keychain-key)# end
R2#
```

Configuring EIGRP Authentication



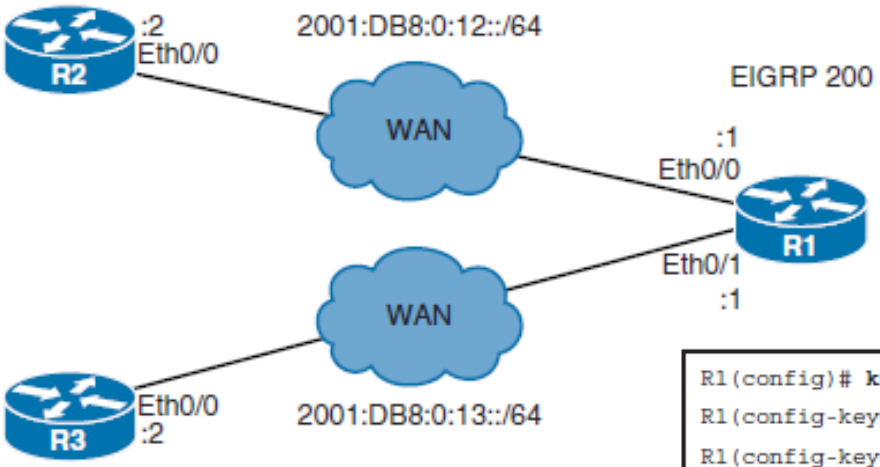

```
R1(config)# interface Ethernet 0/0
R1(config-if)# ip authentication mode eigrp 100 md5
R1(config-if)#
*Sep 20 19:47:43.654: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.10.12.2
(Ethernet0/0) is down: authentication mode changed
R1(config-if)# ip authentication key-chain eigrp 100 EIGRP-KEYS
R1(config-if)#
```

```
R2(config)# interface e0/0
R2(config-if)# ip authentication mode eigrp 100 md5
R2(config-if)# ip authentication key-chain eigrp 100 EIGRP-KEYS
R2(config-if)#
*Sep 20 19:49:56.127: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.10.12.1
(Ethernet0/0) is up: new adjacency
R2(config-if)#
```

Configure EIGRP Key-Based Routing Authentication

```
R1(config)# key chain EIGRP-LIFETIME-KEYS
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string secret-2
R1(config-keychain-key)# accept-lifetime 00:00:00 Jan 1 2014 23:00:00 Mar 20 2015
R1(config-keychain-key)# send-lifetime 00:00:00 Jan 1 2014 23:00:00 Mar 20 2015
R1(config-keychain-key)# key 2
R1(config-keychain-key)# key-string secret-3
R1(config-keychain-key)# accept-lifetime 22:45:00 Mar 20 2015 infinite
R1(config-keychain-key)# send-lifetime 22:45:00 Mar 20 2015 infinite
R1(config-keychain-key)# exit
R1(config)# interface ethernet0/1
R1(config-if)# ip authentication mode eigrp 100 md5
R1(config-if)# ip authentication key-chain eigrp 100 EIGRP-LIFETIME-KEYS
R1(config-if)#
*Sep 20 20:35:13.837: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.10.13.2
(Ethernet0/1) is down: authentication mode changed
R1(config-if)#
```

Configuring EIGRP for IPv6 Authentication



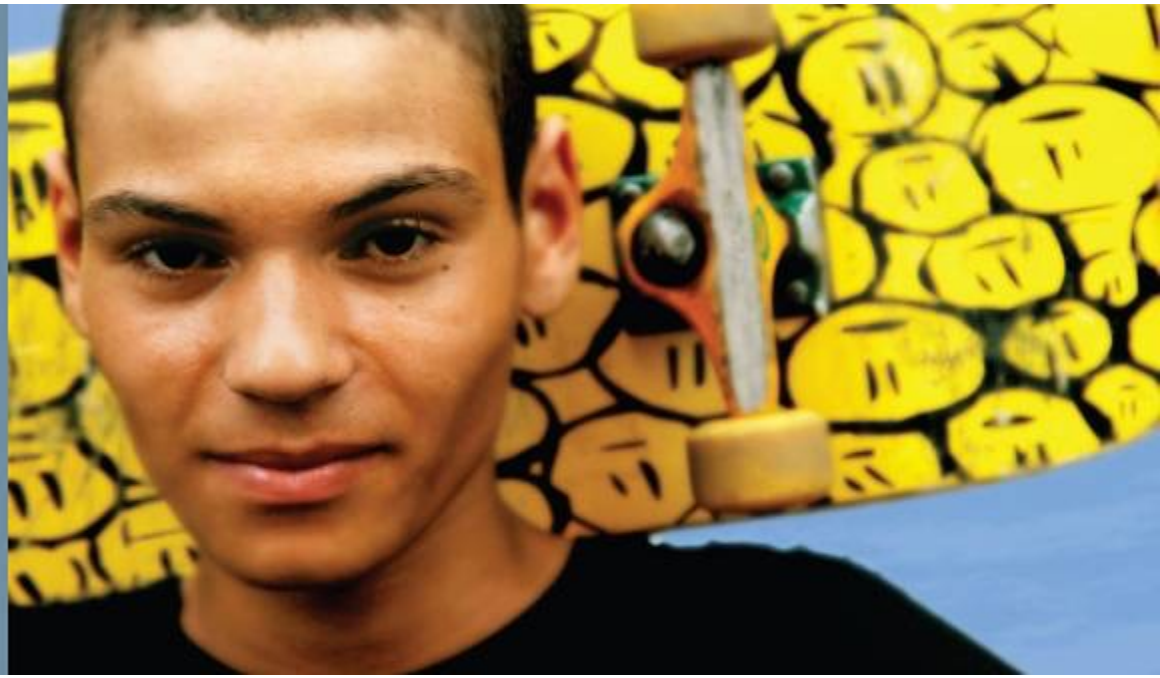

```
R1(config)# key chain R1-IPv6-Chain
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string secret-1
R1(config-keychain-key)# exit
R1(config-keychain)# exit
R1(config)# interface ethernet 0/0
                                          grp 200 md5
                                          GRP-IPv6 200: Neighbor
                                           down: authentication mode changed
                                          in eigrp 200 R1-IPv6-Chain
```

```
R2(config)# key chain R2-IPv6-Chain
R2(config-keychain)# key 1
R2(config-keychain-key)# key-string secret-1
R2(config-keychain-key)# exit
R2(config-keychain)# exit
R2(config)# interface ethernet 0/0
R2(config-if)# ipv6 authentication mode eigrp 200 md5
R2(config-if)# ipv6 authentication key-chain eigrp 200 R2-IPv6-Chain
R2(config-if)# exit
R2(config)# exit
*Sep 20 23:13:09.602: %DUAL-5-NBRCHANGE: EIGRP-IPv6 200: Neighbor
FE80::A8BB:CCFF:FE00:5F00 (Ethernet0/0) is up: new adjacency
R2#
```

Configuring Named EIGRP Authentication

```
R1(config)# key chain NAMED-R1-Chain
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string secret-1
R1(config-keychain-key)# exit
R1(config-keychain)# exit
R1(config)# router eigrp ROUTE
R1(config-router)# address-family ipv4 autonomous-system 110
R1(config-router-af)# network 10.10.0.0 0.0.255.255
R1(config-router-af)# af-interface ethernet 0/0
R1(config-router-af-interface)# authentication key-chain NAMED-R1-Chain
R1(config-router-af-interface)# authentication mode hmac-sha-256 secret-2
R1(config-router-af-interface)# end
R1#
```

Configuring OSPF Authentication



Configuring OSPF Authentication

This section describes how to do the following:

- Configure OSPFv2 neighbor authentication
- Configure OSPFv3 neighbor authentication

OSPF Authentication

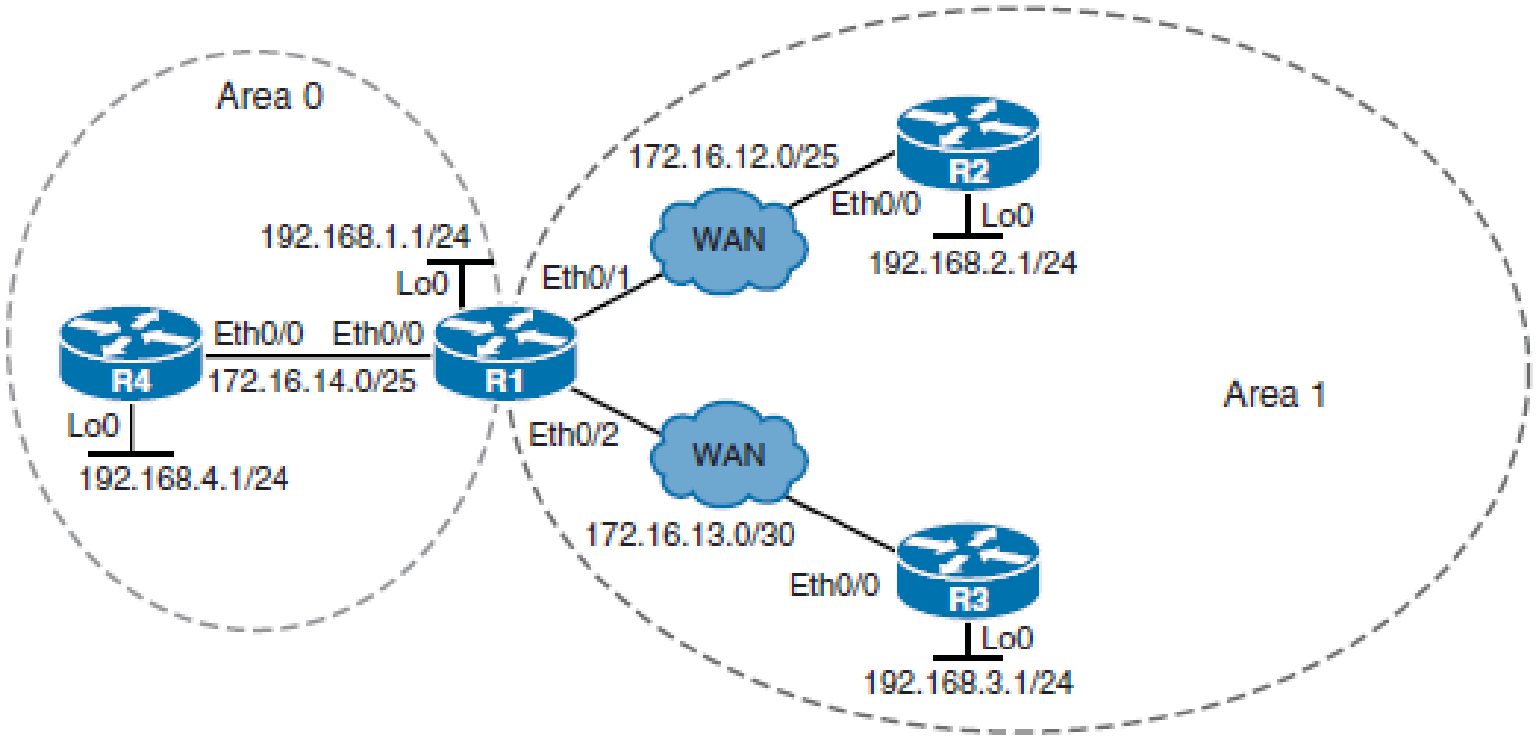
- By default, OSPF does not authenticate routing updates. This means that routing exchanges over a network are not authenticated. OSPFv2 supports
- **Plain-text authentication**
  - Simple password authentication. Least secure and not recommended for production environments.
- **MD5 authentication**
  - Secure and simple to configure using two commands. Should only be implemented if SHA authentication is not supported.
- **SHA authentication**
  - Most secure solution using key chains. Referred to as the OSPFv2 cryptographic authentication feature and only available since IOS 15.4(1)T.

OSPF MD5 Authentication

There are two tasks to enable MD5 hashing authentication:

- **Step 1.**
  - Configure a key ID and keyword (password) using the **ip ospf message-digest key** *key-id* **md5** *password* interface configuration command. The key ID and password are used to generate the hash value that is appended to the OSPF update. The password maximum length is 16 characters. Cisco IOS Software will display a warning if a password longer than 16 characters is entered.
- **Step 2**
  - Enable MD5 authentication using either the **ip ospf authentication message-digest** interface configuration command or the **area** *area-id* **authentication message-digest** OSPF router configuration command. The first command only enables MD5 authentication on a specific interface, and the second command enables authentication for all OSPFv2 interfaces within an area.

Configure OSPF MD5 Authentication



Configure OSPF MD5 Authentication - Interface

```
R1(config)# interface ethernet 0/2
R1(config-if)# ip ospf authentication message-digest
R1(config-if)# ip ospf message-digest-key 1 md5 secret-1
R1(config-if)#
*Sep 21 14:56:55.750: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on Ethernet0/2
from FULL to DOWN, Neighbor Down: Dead timer expired
R1(config-if)#
```

```
R3(config)# interface ethernet 0/0
R3(config-if)# ip ospf authentication message-digest
R3(config-if)# ip ospf message-digest-key 1 md5 secret-1
R3(config-if)#
*Sep 21 14:57:41.473: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Ethernet0/0
from LOADING to FULL, Loading Done
R3(config-if)#
```

Configure OSPF MD5 Authentication in an Area

```
R1(config)# interface ethernet 0/0
R1(config-if)# ip ospf message-digest-key 1 md5 secret-2
R1(config-if)# exit
R1(config)#
R1(config)# router ospf 1
R1(config-router)# area 0 authentication message-digest
R1(config-router)#
*Sep 21 15:22:27.614: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on Ethernet0/0
from FULL to DOWN, Neighbor Down: Dead timer expired
R1(config-router)#
```

```
R4(config)# interface ethernet 0/0
R4(config-if)# ip ospf message-digest-key 1 md5 secret-2
R4(config-if)# exit
R4(config)# router ospf 1
R4(config-router)# area 0 authentication message-digest
R4(config-router)#
*Sep 21 15:23:12.394: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Ethernet0/0 from
LOADING to FULL, Loading Done
R4(config-router)#
```

OSPFv2 Cryptographic Authentication

- **Step 1.**
  - Configure a key chain using the **key chain** *key-name* global configuration command. The key chain contains the key ID and key string and enables the cryptographic authentication feature using the **cryptographic-algorithm** *auth-algo* key chain key configuration mode command.
- **Step 2.**
  - Assign the key chain to the interface using the **ip ospf authentication keychain** *key-name* interface configuration mode command. This also enables the feature.

Configure OSPFv2 Cryptographic Authentication Example

```
R1(config)# key chain SHA-CHAIN
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string secret-1
R1(config-keychain-key)# cryptographic-algorithm ?
  hmac-sha-1    HMAC-SHA-1 authentication algorithm
  hmac-sha-256  HMAC-SHA-256 authentication algorithm
  hmac-sha-384  HMAC-SHA-384 authentication algorithm
  hmac-sha-512  HMAC-SHA-512 authentication algorithm
  md5           MD5 authentication algorithm

R1(config-keychain-key)# cryptographic-algorithm hmac-sha-256
R1(config-keychain-key)# exit
R1(config-keychain)# exit
R1(config-if)# interface s0/0/0
R1(config-if)# ip ospf authentication key-chain SHA-CHAIN
R1(config-if)#
*Sep 21 16:53:03.227: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial0/0/0
from FULL to DOWN, Neighbor Down: Dead timer expired
R1(config-if)#
```

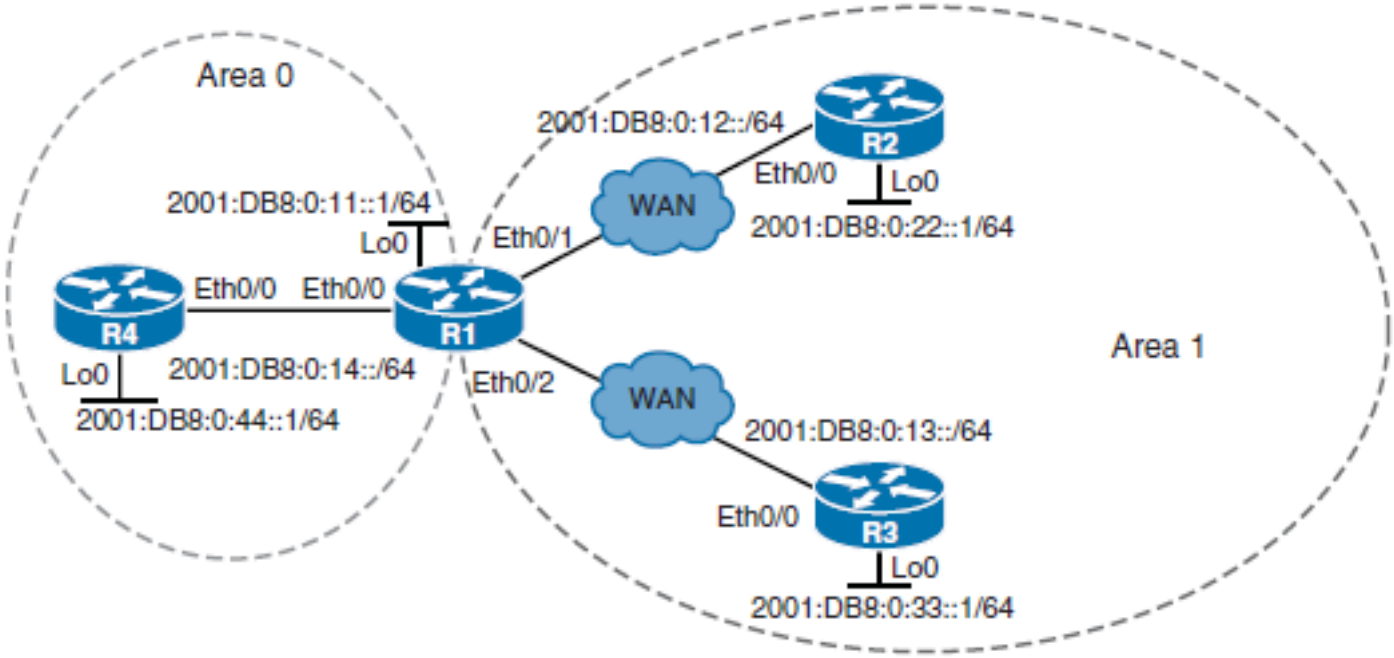
OSPFv3 Authentication

- OSPFv3 requires the use of IPsec to enable authentication.
- In OSPFv3, authentication fields have been removed from OSPFv3 packet headers.
- When OSPFv3 runs on IPv6, OSPFv3 requires the IPv6 Authentication Header (AH) or IPv6 Encapsulating Security Payload (ESP) header to ensure integrity, authentication, and confidentiality of routing exchanges.
- To deploy OSPFv3 authentication, first define the security policy on each of the devices within the group. The security policy consists of the combination of the key and the security parameter index (SPI). The SPI is an identification tag added to the IPsec header.

Configuring OSPFv3 Authentication

- The authentication policy can be configured either on an
- **Interface**
  - Can be configured using either the **ospfv3 authentication** { **ipsec** *spi* } { **md5** | **sha1** } { *key-encryption-type key* } | **null** interface configuration command or the **ipv6 ospf authentication** { **null** | **ipsec spi** *spi authentication-algorithm* [*keyencryption-type*] [*key*]} interface configuration commands. A *key* with the key length of exactly 40 hex characters must be specified.
- **Area**
  - Use the **area** *area-id* **authentication ipsec spi** *spi authentication-algorithm* [*key-encryption-type*] *key* router configuration mode. When configured for an area, the security policy is applied to all the interfaces in the area. For higher security, use a different policy on each interface.

Configuring OSPFv3 Authentication on an Interface Example



Configuring OSPFv3 Authentication on an Interface Example

```
R1(config)# interface Ethernet0/1
R1(config-if)# ipv6 ospf authentication ipsec spi 300 sha1
1234567890123456789012345678901234567890
R1(config-if)#
*Sep 21 19:56:02.195: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R1(config-if)#
*Sep 21 19:56:35.245: %OSPFv3-5-ADJCHG: Process 1, IPv6, Nbr 2.2.2.2 on
Ethernet0/1 from FULL to DOWN, Neighbor Down: Dead timer expired
R1(config-if)#
```

```
R2(config)# interface Ethernet 0/0
R2(config-if)# ipv6 ospf authentication ipsec spi 300 sha1 12345678901234567890123456
78901234567890
R2(config-if)#
*Sep 21 19:58:51.543: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R2(config-if)#
*Sep 21 19:58:55.179: %OSPFv3-5-ADJCHG: Process 1, IPv6, Nbr 1.1.1.1
on Ethernet0/0 from LOADING to FULL, Loading Done
R2(config-if)#
```

Configuring OSPFv3 Authentication for Area 0

```
R1(config)# router ospfv3 1
R1(config-router)# area 0 authentication ipsec spi 500 sha1 12345678901234567890123456789012345678901234567890
R1(config-router)#
*Sep 21 20:02:24.415: %OSPFv3-5-ADJCHG: Process 1, IPv6, Nbr 4.4.4.4 on Ethernet0/0 from FULL to DOWN, Neighbor Down: Dead timer expired
R1(config-router)#
```

```
R4(config)# router ospfv3 1
R4(config-router)# area 0 authentication ipsec spi 500 sha1 12345678901234567890123456789012345678901234567890
R4(config-router)#
*Sep 21 20:02:29.367: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R4(config-router)#
*Sep 21 20:02:31.186: %OSPFv3-5-ADJCHG: Process 1, IPv6, Nbr 1.1.1.1 on Ethernet0/0 from LOADING to FULL, Loading Done
R4(config-router)#
```

Configuring BGP Authentication

Configuring BGP Authentication

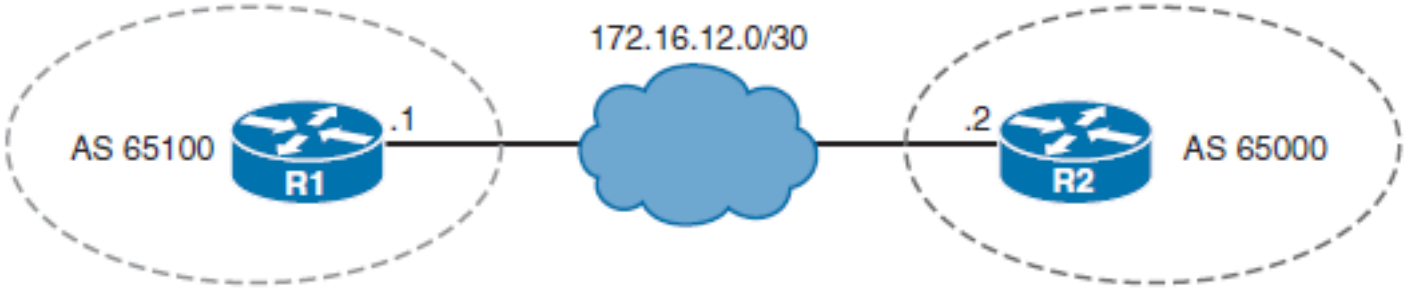
This section covers the following topics:

- How BGP authentication using MD5 hashes works
- Configuring and verifying BGP for IPv4 authentication
- Configuring and verifying BGP for IPv6 authentication

BGP Authentication Configuration Checklist

- BGP neighbor authentication can be configured on a router so that the router authenticates the source of each routing update packet that it receives. This authentication is accomplished by the exchange of an authentication key.
- Like EIGRP and OSPF, BGP also supports MD5 neighbor authentication. To generate an MD5 hash value, BGP uses the shared secret key and portions of the IP and TCP headers and the TCP payload.
- The MD5 hash is then stored in TCP option 19, which is created specifically for this purpose by RFC 2385.
- Successful MD5 authentication requires the same password on both BGP peers.
- Configuring MD5 authentication causes Cisco IOS Software to generate and check the MD5 digest of every segment that is sent on the TCP connection.

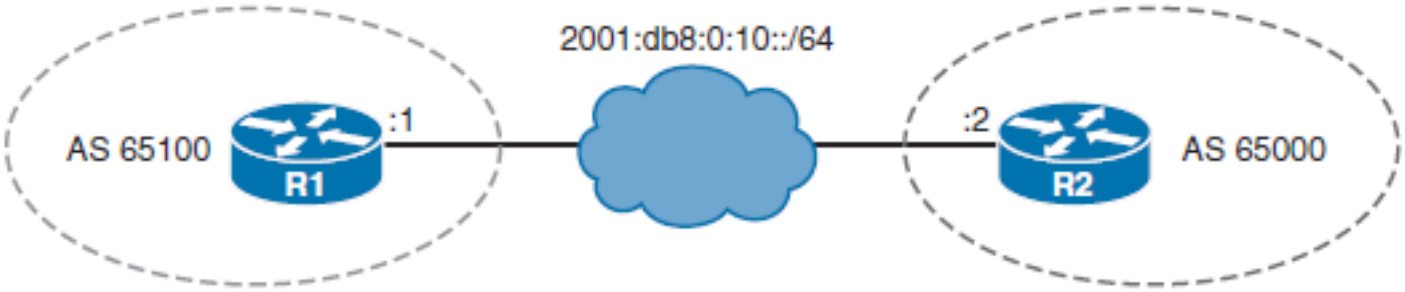
BGP Authentication Configuration




```
R1(config)# router bgp 65100
R1(config-router)# neighbor 172.16.12.2 remote-as 65000
R1(config-router)# neighbor 172.16.12.2 password secret-1
R1(config-router)#
```

```
R2(config)# router bgp 65000
R2(config-router)# neighbor 172.16.12.1 remote-as 65100
R2(config-router)# neighbor 172.16.12.1 password secret-1
R2(config-router)#
```

BGP for IPv6 Authentication Configuration




```
R1(config)# router bgp 65100
R1(config-router)# neighbor 2001:db8:0:10::2 remote-as 65000
R1(config-router)# neighbor 2001:db8:0:10::2 password secret-2
R1(config-router)#
```

```
R2(config)# router bgp 65000
R2(config-router)# neighbor 2001:db8:0:10::1 remote-as 65100
R2(config-router)# neighbor 2001:db8:0:10::1 password secret-2
R2(config-router)#
```

Implementing VRF-Lite

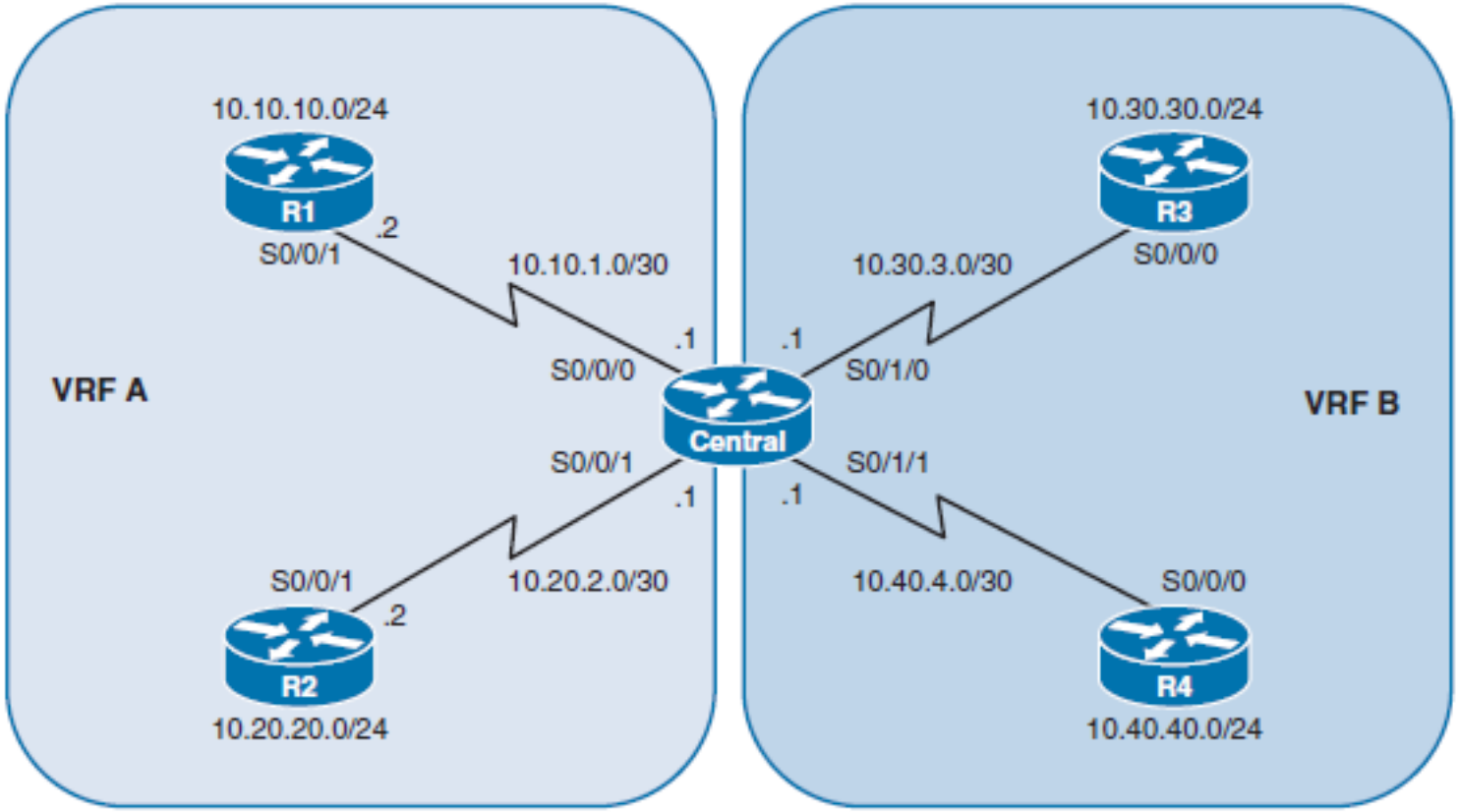
Implementing VRF-Lite

- Virtual Routing and Forwarding (VRF) is a technology that allows the device to have
- multiple but separate instances of routing tables exist and work simultaneously.
- A VRF instance is essentially a logical router and consists of an IP routing table, a forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.
- A VRF increases
  - Network functionality by allowing network paths to be completely segmented without using multiple devices.
  - Network security because traffic is automatically segmented. VRF is conceptually similar to creating Layer 2 VLANs but operates at Layer 3.
- Service providers (SPs) often take advantage of VRF to create separate virtual private networks (VPNs) for customers. Therefore, VRF is often referred to as *VPN routing and forwarding* .

VRF and VRF-Lite

- VRF is usually associated with a service provider running Multiprotocol Label Switching (MPLS) because the two work well together. In a provider network, MPLS isolates each customer's network traffic, and a VRF is maintained for each customer.
- However, VRF can be used in other deployments without using MPLS.
- VRF-lite is the deployment of VRF without MPLS. With the VRF-lite feature, the Catalyst switch supports multiple VPN routing/forwarding instances in customer-edge devices.
- VRF-lite allows an SP to support two or more VPNs with overlapping IP addresses using one interface. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF.
- Interfaces in a VRF can be either physical, such as Ethernet or serial ports, or logical, such as VLAN SVIs. However, a Layer 3 interface cannot belong to more than one VRF at any time.

Enabling VRF



Enabling VRF

```
Central(config)# ip vrf VRF-A
Central(config-vrf)# exit
Central(config)# ip vrf VRF-B
Central(config-vrf)# exit
Central(config)# interface Serial0/0/0
Central(config-if)# ip vrf forwarding VRF-A
Central(config-if)# ip address 10.10.1.1 255.255.255.252
Central(config-if)# clock rate 2000000
Central(config-if)# no shut
Central(config-if)# exit
Central(config)#
Central(config-if)# interface Serial0/0/1
Central(config-if)# ip vrf forwarding VRF-A
Central(config-if)# ip address 10.20.2.1 255.255.255.252
Central(config-if)# no shut
Central(config-if)# exit
Central(config)#
Central(config-if)# interface Serial0/1/0
Central(config-if)# ip vrf forwarding VRF-B
Central(config-if)# ip address 10.30.3.1 255.255.255.252
Central(config-if)# clock rate 2000000
Central(config-if)# no shut
Central(config-if)# exit
Central(config)#
Central(config-if)# interface Serial0/1/1
Central(config-if)# ip vrf forwarding VRF-B
Central(config-if)# ip address 10.40.4.1 255.255.255.252
Central(config-if)# no shut
Central(config-if)# exit
Central(config)#
```

Verify the Routing Table in VRF Environment

```
Central# show ip route | begin Gateway
Gateway of last resort is not set

Central#
Central# show ip route vrf VRF-A | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.10.1.0/30 is directly connected, Serial0/0/0
L        10.10.1.1/32 is directly connected, Serial0/0/0
C        10.20.2.0/30 is directly connected, Serial0/0/1
L        10.20.2.1/32 is directly connected, Serial0/0/1
Central#
Central# show ip route vrf VRF-B | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.30.3.0/30 is directly connected, Serial0/1/0
L        10.30.3.1/32 is directly connected, Serial0/1/0
C        10.40.4.0/30 is directly connected, Serial0/1/1
L        10.40.4.1/32 is directly connected, Serial0/1/1
Central#
```

Enable EIGRP for VRF-A

```
Central(config)# router eigrp 1
Central(config-router)# address-family ipv4 vrf VRF-A
Central(config-router-af)# network 10.10.1.0 0.0.0.3
Central(config-router-af)# network 10.20.2.0 0.0.0.3
Central(config-router-af)# autonomous-system 1
Central(config-router-af)# no auto-summary
Central(config-router-af)#
*Aug  5 04:45:35.879: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.20.2.2
(Serial0/0/1) is up: new adjacency
*Aug  5 04:45:35.883: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.10.1.2
(Serial0/0/0) is up: new adjacency
Central(config-router-af)# ^Z
Central#
```

Verify the Routing Table of VRF-A

```
Central# show ip route vrf VRF-A | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C        10.10.1.0/30 is directly connected, Serial0/0/0
L        10.10.1.1/32 is directly connected, Serial0/0/0
D        10.10.10.0/24 [90/2297856] via 10.10.1.2, 00:00:06, Serial0/0/0
C        10.20.2.0/30 is directly connected, Serial0/0/1
L        10.20.2.1/32 is directly connected, Serial0/0/1
D        10.20.20.0/24 [90/2297856] via 10.20.2.2, 00:05:41, Serial0/0/1
Central# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
% No usable Router-ID found
Central#
Central# show ip eigrp vrf VRF-A neighbors
EIGRP-IPv4 Neighbors for AS(1) VRF(VRF-A)
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   10.20.2.2               Se0/0/1           13 00:43:42    3   100  0  4
0   10.10.1.2               Se0/0/0           11 00:47:54    1   100  0  5
Central#
```

Enable OSPF for VRF-B

```
Central(config)# router ospf 1 vrf VRF-B
Central(config-router)# router-id 5.5.5.5
Central(config-router)# network 10.30.3.0 0.0.0.3 area 0
Central(config-router)# network 10.40.4.0 0.0.0.3 area 0
Central(config-router)#
*Aug  5 04:47:22.327: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on Serial0/1/0 from
LOADING to FULL, Loading Done
*Aug  5 04:47:22.467: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on Serial0/1/1 from
LOADING to FULL, Loading Done
Central(config-router)# ^Z
Central#
```

Verify the Routing Table of VRF-B

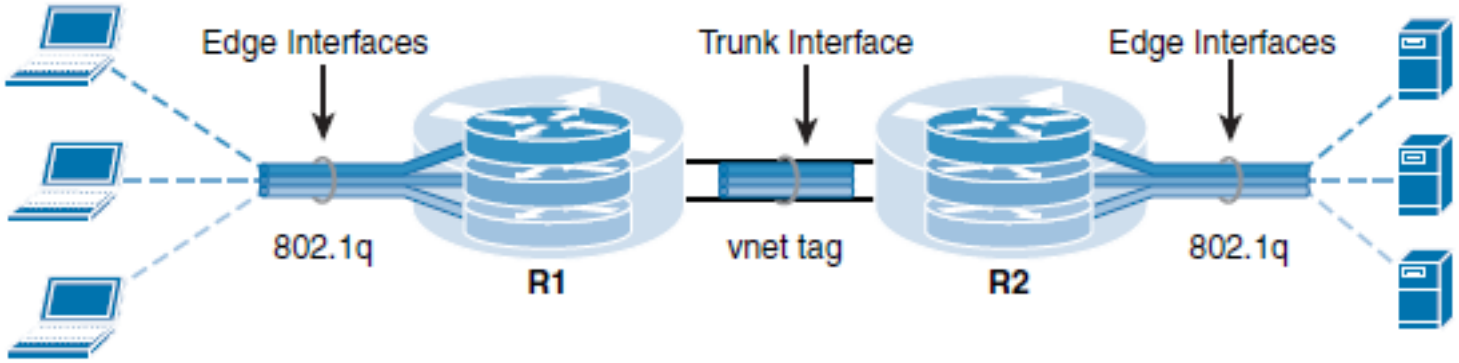
```
Central# show ip route vrf VRF-B | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C        10.30.3.0/30 is directly connected, Serial0/1/0
L        10.30.3.1/32 is directly connected, Serial0/1/0
O        10.30.30.0/24 [110/65] via 10.30.3.2, 00:05:07, Serial0/1/0
C        10.40.4.0/30 is directly connected, Serial0/1/1
L        10.40.4.1/32 is directly connected, Serial0/1/1
O        10.40.40.0/24 [110/65] via 10.40.4.2, 00:07:30, Serial0/1/1
Central#
```

Easy Virtual Network

- For true path isolation, Cisco Easy Virtual Network (EVN) provides the simplicity of Layer 2 with the controls of Layer 3.
- EVN provides traffic separation and path isolation capabilities on a shared network infrastructure.
- EVN is an IP-based network virtualization solution that takes advantage of existing VRFlite technology to:
  - Simplify Layer 3 network virtualization
  - Improve support for shared services
  - Enhance management and troubleshooting

Easy Virtual Network

- EVN reduces network virtualization configuration significantly across the entire network infrastructure by creating a virtual network trunk. The traditional VRF-lite solution requires creating one subinterface per VRF on all switches and routers involved in the data path, creating a lot of burden in configuration management.
- EVN removes the need of per-VRF subinterface by using the **`vnet trunk`** interface command.



Chapter 8 Summary

- Write and follow a security policy before securing a device.
- Passwords are stored in the configuration and should be protected from eavesdropping.
- Use SSH instead of Telnet, especially when using it over an unsecure network.
- Create router ALCs to protect the infrastructure by filtering traffic on the network edge.
- Secure SNMP if it is used on the network.
- Periodically save the configuration in case it gets corrupted or changed.
- Implement logging to an external destination to have insight into what is going on in a network.
- Disable unused services.
- Unauthorized routers might launch a fictitious routing update to convince a router to send traffic to an incorrect destination. Routers authenticate the source of each routing update that is received when routing authentication is enabled.
- There are two types of routing authentication: plain-text and hashing authentication.

Chapter 8 Summary

- Avoid using plain-text authentication.
- A key chain is a set of keys that can be used with routing protocol authentications.
- Different routing protocols support different authentication options.
- When EIGRP authentication is configured, the router verifies every EIGRP packet.
- Classic EIGRP for IPv4 and IPv6 supports MD5 authentication, and named EIGRP supports SHA authentication.
- To configure classic MD5 authentication, define a key, enable EIGRP authentication mode on the interface, and associate the configured key with the interface.
- To configure SHA authentication, you need to use EIGRP named configuration mode.

Chapter 8 Summary

- Verify the EIGRP authentication by verifying neighborship.
- When authentication is configured, the router generates and checks every OSPF packet and authenticates the source of each update packet that it receives.
- In OSPFv2 simple password authentication the routers send the key that is embedded in the OSPF packets.
- In OSPFv2 MD5 authentication the routers generate a hash of the key, key ID, and message. The message digest is sent with the packet.
- OSPFv3 uses native functionality offered by IPv6. All that is required for OSPFv3 authentication is IPsec AH. AH provides authentication and integrity check. Ipsec ESP provides encryption for payloads, which is not required for authentication.
- BGP authentication uses MD5 authentication.
- Router generates and verifies MD5 digest of every segment sent over the BGP connection.
- Verify BGP authentication by verifying if BGP sessions are up.

Chapter 8 Labs

- **CCNPv7 ROUTE Lab8.1 Secure Management Plane**
- **CCNPv7 ROUTE Lab8.2 Routing Protocol Authentication**

Acknowledgment

- Some of images and texts are from Implementing Cisco IP Routing (ROUTE) Foundation Learning Guide by Diane Teare, Bob Vachon and Rick Graziani (1587204568)

- Special Thanks to *Bruno Silva*