Copyright Information

Disclaimer

The following publication, CCIE R&S Lab Workbook Volume I Version 5.0, is designed to assist candidates in the preparation for Cisco Systems' CCIE Routing & Switching Lab Exam. While every effort has been made to ensure that all material is as complete and accurate as possible, the enclosed material is presented on an "as is" basis. Neither the authors nor Internetwork Expert, Inc. assume any liability or responsibility to any person or entity with respect to loss or damages incurred from the information contained in this workbook.

This workbook was developed by Internetwork Expert, Inc. and is an original work of the aforementioned authors. Any similarities between material presented in this workbook and actual CCIE lab material is completely coincidental.

Table of Contents

System Management

> **Note**
>
> Erase and reload all devices and load the *System Management* initial configurations prior to starting.

12.1 Exec Aliases

- Configure aliases on R1 to accomplish this following:
  - Issuing the command **`ri`** in exec mode should view the routing table contents, while **`rb`** views the BGP table.
  - Issuing **`i Fa0/0`** in global configuration should switch to interface FastEthernet 0/0 configuration mode.
  - Issuing **`iea ACL`** in global configuration should create the extended access-list named ACL.
  - Issuing **`so`** and **`si`** should apply a policy-map to an interface either outbound or inbound respectively.

12.2 System Message Logging

- Enable system message logging on R4 and R6 as follows:
  - Both routers should save debugging messages to their internal buffers up to 8192 bytes.
  - Debugging messages should be sent to the router consoles, but limited to 1 message per second.
  - Console log messages should not interrupt other command output.
  - Users logged via telnet should only see informational level messages and above.

12.3 Syslog Logging

- Configure R4 and R6 to log messages to syslog as follows:
  - Both R4 and R6 should log to the server 155.X.146.100.
  - Log all messages up to notifications.
  - R4 should use reliable transport at port 5000.
  - R6 should use the default transport.
  - Use IDs ROUTER4 and ROUTER6 respectively and the UNIX facility LOCAL1.
  - Messages should be sourced off of the routers' Loopback0 interfaces.
  - Set the message queue depth to 256.

12.4 Logging Counting and Timestamps

- Configure R4 and R6 to count error messages.
- R4 should insert uptime-based timestamps in log messages, while R6 should insert date-time based timestamps.
- R6 should add milliseconds based timestamps to debugging messages only.
- Include year information in logging messages on R6 but not in debugging messages.
- Provide a solution to prevent against tampering with stored syslog information.

12.5 Logging to Flash Memory

- Configure SW1 to log notification messages in its internal flash memory.
- Create the directory /var/log and store messages in the file named "syslog".
- Set the maximum file size to 32768 bytes.

12.6 Configuration Change Notification and Logging

- Configure R4 to locally track changes made to its running configuration.
- Log these changes via syslog, but ensure passwords in the configuration will not be sent across this communication channel.
- Set the queue size to 1000.

12.7 Configuration Archive & Rollback

- Configure SW1 to store its configuration archive on the TFTP server 155.X.58.100 using “sw1-config” as the prefix.
- Archive the configuration every 24 hours.
- Additionally the configuration should be archived every time the running config is saved to NVRAM.

12.8 Logging with Access-Lists

- Configure R4 to log every RIP packet entering the interface connected to VLAN 146.
- Generate a cumulative log entry for every 2 matched packets.
- Send a logging message no more than once per 10 seconds.
- Log the MAC address of the device sending the packet.

12.9 TCP Keepalives

- Configure R1 so that it’s capable of detecting dead TCP connections and closing them prematurely.

12.10 Generating Exception Core Dumps

- Configure R3 to save core dumps to an FTP server at 155.X.146.100 under the name “r3-core”.
- Use active FTP and the username/password values cisco/cisco.
- Configure the router to generate a memory dump and reload as soon as free memory falls below 1Mbyte.
- The router should also reload in the case that memory fragmentation prohibits a process from allocating more than 64Kbytes of memory.
- Disable the local crash information collection.

12.11 Conditional Debugging

- Configure R6 to debug RIP updates entering and leaving the connection to VLAN 67 only.

12.12 Telnet Service Options

- Configure R3 to source all telnet sessions from its Loopback0 interface, and to use a marking of IP Precedence 3 for these packets.
- Do not display any telnet informational messages.
- Create a hostname entry “R4” that pointing to the IP address of R4.
- Ensure that a user typing “R4” in exec mode is connected without seeing the IP address of R4.
- Idle outgoing telnet sessions should signal the remote host to pause output.
- Display the message “Sorry, your connection failed” when a telnet connection to the above host fails.

12.13 Tuning Packet Buffers

- Enable automatic buffer tuning on R3.
- Configure R4 to double the permanent number of all public (not interface specific) buffers.
- Ensure there are at least 10 Large and Huge buffers always available.

12.14 Terminal Line Settings

- Configure R3 to allow only telnet connections to VTY lines 0 through 4 without using an access-list.
- When a user is telnetted into R3 and mistypes a command in exec mode, R3 should not try to open a telnet session to the command as if it was a hostname.
- Configure VTY line 0 to listen for telnet at port 3001.
- When the virtual terminal line is busy issue the output “Sorry, the line is already in use” to the connecting user.
- Exec sessions on a VTY line should timeout after 2 minutes of inactivity; additionally a user should not be able to hold the line busy for more than 5 minutes.
- Sessions initiated from a VTY line should time out in 1 minute.
- When the console line is idle the user should see the output “Welcome to IOS”
- Allow no more than 1 session to be initiated from the console line.
- The terminal length should be no more than 20 lines.
- IP netmasks should be displayed using hex numbers.
- Allow a user to lock VTY terminal lines.

12.15 SNMPv2c Server

- Configure SNMP on R4 with the SNMP contact set to “Default Contact” and the SNMP Location to “Default Location”.
- Enable the SNMP server service using the community string value of CISCO so that remote hosts can modify the local MIB database.
- Allow the system to be reloaded via SNMP.
- Ensure that interface index numbers persist between reloads.
- Only allow configuration transfers via TFTP to/from the host 155.X.146.100.

12.16 SNMPv2c Access Control

- Configure R4 to restrict read-write access to only hosts in VLAN 146.
- Log any other SNMP access attempts using the community string “CISCO”.
- Allow any host to access R4’s MIBs in read-only mode using the community string “PUBLIC”.
- Read-only access should be permitted to the “cisco” MIB subtree only.

12.17 SNMP Traps and Informs

- Configure R4 to send SNMP traps to the management host 155.X.146.100 and informs to the host 155.X.146.101.
- Only send notifications about interfaces going up or down.
- Ensure no informs are being sent about Serial interfaces changing their status.
- Use the community value of “CISCO” with the above traps and informs.

12.18 CPU and Memory Thresholds

- Configure R4 to monitor CPU usage every 5 seconds using the **`process cpu`** command, and to generate a rising threshold event every time the CPU usage hits 50%.
- Set up the free memory low threshold to 1000Kbytes, and reserve 512Kbytes of memory for the notification process using the **`memory`** command.

12.19 SNMPv3

- Create two SNMP views named NORMAL and RESTRICTED on R6.
- The NORMAL view should include the branch “iso”.
- The RESTRICTED view should include the branch “ifEntry.*.*n*”, where *n* is the ifIndex of R6’s serial interface.
- Create an SNMP group named NORMAL with the read/write view assigned to NORMAL, and a security model of “priv”.
- Assign the user named NORMAL to this group, set the SHA1 password to CISCO, and the encryption key to CISCO.
- Create an SNMP group named RESTRICTED with the read view assigned to RESTRICTED, and a security model of “auth”.
- Only users in VLAN 146 should be allowed in the SNMP group RESTRICTED.
- Assign the user named RESTRICTED to this group using the security model which only requires authentication using the password CISCO.
- Create an SNMP group named TRAP with the security model “priv”.
- Assign the user named TRAP to this group using the password and encryption key CISCO.
- Enable SNMP traps for LinkUp and LinkDown events only, and send them to the destination host 155.X.146.100 using the security model “priv” and the username TRAP.

12.20 SNMP MAC Address Notifications

- Configure SW1 to notify the NMS station at the IP address 155.X.146.100 about MAC addresses learned on ports connected to R1 and R5.
- Use the community value of CISCO to send notifications.
- Limit the rate of notifications to 1 per second.
- Report both added and removed MAC addresses.
- Store the latest 100 notification events in history buffer.

12.21 SNMP Notifications of Syslog Messages

- Configure R1 so that all debugging and higher priority level messages are sent via SNMP to an NMS host at the IP address 155.X.146.100.
- Set the syslog to SNMP buffer size to 100 messages.

12.22 CDP

- Configure R4 to send CDP announcement every 10 seconds, and instruct other devices to hold the updates for 40 seconds.
- Use the Loopback0 interface for IP address advertisements in CDP messages.
- Disable logging of duplex mismatch detected via CDP messages.
- Do not send or receive CDP message on R4's connection to VLAN 43.

12.23 RMON Alarms

- Configure R1 to monitor the rate of packets entering its interface connecting to VLAN 146 based on the total input packet counter (ifEntry.11).
- If the rate is above 100 packets per minute generate the log message "VLAN146 Interface Congested" and send a trap to the management host at 155.X.146.100.
- Use the SNMP community string CISCO.
- When the packet rate falls below 50 packets per minute, generate another log message "VLAN146 Interface UnCongested" and send a corresponding trap.
- The owner of events should be set to "CISCO".

12.24 RMON Statistics Collection

- Configure SW1 to collect RMON statistics on packets entering the interface connected to R3.
- Store historical information for the last 100 samples of packet statistics taken every 30 seconds.

12.25 HTTP Server and Client

- Configure R4 to support the transfer of the IOS image stored in R6's flash using HTTP.
- R4 should authenticate with a username/password pair of CISCO/CISCO stored in the local database of R6
- R6 should only allow HTTP connections at port 8080, from the subnet 150.X.0.0/16, and limit the maximum number of concurrent connections to two.
- Change the default WWW port to 8080.
- Allow R4 to use secure HTTP connection on port 4043, but limit the security features to use only the DES symmetric cipher.

12.26 FTP Server and Client

- Configure the FTP server service on R6 to serve IOS images off of flash.
- R4 should initiate an FTP session to R6 with the username/password CISCO/CISCO.
- When a transfer occurs R6 should initiate the FTP data channel back to the client.
- Source FTP connections off R4 Loopback0 interface

12.27 TFTP Server and Client

- Configure the TFTP server service on R6.
- When someone requests the file “r6-ios” from R6 it should return its IOS image file.
- Ensure only R1’s Loopback0 interface is allowed to connect and retrieve files via TFTP from R6.
- Configure R1 to source TFTP packets off its Loopback0 interface.

12.28 Remote Shell

- Configure the network so that R1 is to view the running configuration of R6 using remote shell.
- Source remote-shell connections off the Loopback0 interface of R1
- Allow RCP file-copy from R6 to R1.

12.29 NTP

- Configure R4 and R6 as authoritative NTP time sources with stratum 5.
- Both R4 and R6 should synchronize with each other.
- R5 should poll R4 and R6 for time updates, but prefer R4.
- Configure R5 to broadcast NTP updates on its connection to VLAN 58, and for SW2 to listen to NTP broadcasts on this link.
- Configure R6 to send multicast NTP packets to the address 239.1.1.1 on its connection to VLAN 67.
- SW1 and SW3 should be able to synchronize with R6 using these multicast packets.

12.30 NTP Authentication

- Configure R4 and R6 to authenticate each others peering session using the key “CISCO46”.
- R5 should authenticate NTP packets received from R4 and R6 using key values “CISCO4” and “CISCO6” respectively.
- SW2 should only accept NTP updates on VLAN58 if they are authenticated using key value of “CISCO58”.

12.31 NTP Access Control

- Configure R4 and R6 so that no other devices can update their clocks.
- R5 should only allow R4 and R6 to update its clock.
- R4 and R6 should not serve time out to other devices besides R5.
- SW2 should only accept time from R5 and not other sources, but R5 should be allowed to serve time out to other devices.

12.32 Auto-Install over LAN Interfaces using DHCP

- Configure the network to support auto-install as follows:
    - Upon booting R4 should request an IP address for its FastEthernet0/1 interface via DHCP.
    - R1 should forward this request onto R5.
    - R5 should reply with the IP address 155.X.146.4/24, the default gateway 155.X.146.1, the DNS server 155.X.146.6, and the TFTP server “RackXSW2”.
    - R4 should then ask R6 what its own hostname is, and download the config file “RackXR4-confg” from SW2 via TFTP.

12.33 Auto-Install over Frame-Relay

- Disable R1’s link to VLAN 146.
- Configure R5 as a staging router for AutoInstall over Frame-Relay for R4 as follows:
    - R5 should assign R4 the IP address 155.X.0.4.
    - When R5 receives a TFTP request from R4 for the file “network-confg” it should be forwarded to SW2.
    - R4 should download this config from SW2 and learn its hostname RackXR4.
    - R4 should then download the config file “RackXR4-confg” from SW2.

12.34 Auto-Install over LAN Interfaces using RARP

- Configure R1 to supply an IP address to R4 via RARP on VLAN 146 when it boots.
- SW2 should supply the network and host-specific configuration files to R4.

12.35 IOS Menus

- When a user named OPERATOR telnets into R1 and authenticates with the password of CISCO it should see the following menu:

```
Operator Menu


    1           Display IP Routing Table

    2           Display Running Config

    3           Escape to Shell

    4           Disconnect
```

- Option 1 should show the routing table.
- Option 2 should show the running configuration.
- Option 3 should exit to the exec prompt.
- Option 4 should exit the telnet session.
- Clear the screen when the menu starts.

12.36 IOS Banners

- Change the default IOS banners on R3 as follows:
  - All connecting users, with the exception to console users, should see the notification message “Welcome to IOS Router”.
  - Before the login prompt is displayed users should see the message “Please, authenticate yourself”.
  - Before a user can see the shell prompt, display “Hi, you are on the line <Line> have a nice time at <Hostname>” where <Line> is the line number the user is connected to and <Hostname> is the current router’s hostname.
  - Set the banner for reverse telnet connections to “This is a reverse telnet connection”.

12.37 KRON Command Schedule

- Configure a KRON occurrence on R4 that saves the running config to a remote TFTP server daily at 08:00.
- Use the server 155.1.146.100 and the filename “r3-config”

System Management Solutions

12.1 Exec Aliases

- Configure aliases on R1 to accomplish this following:
  - Issuing the command **ri** in exec mode should view the routing table contents, while **rb** views the BGP table.
  - Issuing **i Fa0/0** in global configuration should switch to interface FastEthernet 0/0 configuration mode.
  - Issuing **iea ACL** in global configuration should create the extended access-list named ACL.
  - Issuing **so** and **si** should apply a policy-map to an interface either outbound or inbound respectively.

Configuration

```
R1:
alias interface si service-policy input
alias interface so service-policy output
alias configure i interface
alias configure iae ip access-list extended
alias exec ri show ip route
alias exec rb show ip bgp
```

Verification

> **Note**
>
> Command aliases in IOS can be used as shortcuts to speed up and simplify repetitive configurations. Aliases can be configured for other parser modes, such as route-map mode or class-map mode, but the most common ones are exec, global configuration, and interface mode. These configurations can be verified as follows:

```
Rack1R1(config)#i fastEthernet 0/0
Rack1R1(config-if)#exit

Rack1R1(config)#iae ACL
Rack1R1(config-ext-nacl)#exit
Rack1R1(config)#iae 100
Rack1R1(config-ext-nacl)#exit
```



```
Rack1R1(config)#i fastEthernet 0/0
Rack1R1(config-if)#so TEST
% policy map TEST not configured
Rack1R1(config-if)#si TEST
% policy map TEST not configured

Rack1R1#ri
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R    222.22.2.0/24 [120/9] via 155.1.13.3, 00:00:03, Serial0/1
                   [120/9] via 155.1.0.2, 00:00:03, Serial0/0
R    204.12.1.0/24 [120/1] via 155.1.146.4, 00:00:00, FastEthernet0/0
     155.1.0.0/24 is subnetted, 15 subnets
C       155.1.146.0 is directly connected, FastEthernet0/0
R       155.1.23.0 [120/1] via 155.1.13.3, 00:00:03, Serial0/1
R       155.1.10.0 [120/3] via 155.1.0.5, 00:00:03, Serial0/0
R       155.1.8.0 [120/2] via 155.1.0.5, 00:00:03, Serial0/0
R       155.1.9.0 [120/3] via 155.1.146.6, 00:00:04, FastEthernet0/0
C       155.1.13.0 is directly connected, Serial0/1
C       155.1.0.0 is directly connected, Serial0/0
R       155.1.7.0 [120/2] via 155.1.146.6, 00:00:04, FastEthernet0/0
R       155.1.5.0 [120/1] via 155.1.0.5, 00:00:04, Serial0/0


Rack1R1#ri 222.22.2.0
Routing entry for 222.22.2.0/24
  Known via "rip", distance 120, metric 9
  Redistributing via rip
  Last update from 155.1.13.3 on Serial0/1, 00:00:03 ago
  Routing Descriptor Blocks:
  * 155.1.13.3, from 155.1.13.3, 00:00:03 ago, via Serial0/1
      Route metric is 9, traffic share count is 1
    155.1.0.2, from 155.1.0.5, 00:00:05 ago, via Serial0/0
      Route metric is 9, traffic share count is 1

Rack1R1#rb
% BGP not active
```


12.2 System Message Logging

- Enable system message logging on R4 and R6 as follows:
  - Both routers should save debugging messages to their internal buffers up to 8192 bytes.
  - Debugging messages should be sent to the router consoles, but limited to 1 message per second.
  - Console log messages should not interrupt other command output.
  - Users logged via telnet should only see informational level messages and above.

Configuration

```
R4:
logging on
logging buffered 8192 debugging
logging console debugging
logging rate-limit console all 1
logging monitor informational
!
line console 0
 logging synchronous

R6:
logging on
logging buffered 8192 debugging
logging console debugging
logging rate-limit console all 1
logging monitor informational
!
line console 0
 logging synchronous
```

Verification

Note

Logging messages can be sent to multiple destinations simultaneously, including the console/terminal lines, the internal buffer, and a remote syslog server. Some platforms also allow storing logging messages in the local flash memory.

For each destination you can configure the message severity level, which controls which messages are logged. Logging at severity 7 (debugging) includes all messages 0 through 7. Logging at severity 4 includes all messages 0 through 4.

For syslog the logging facility is also available, which controls the format of the log message. Using different facilities for different devices, for example local5 for all routers and local6 for all switches, can make sorting through log files on the syslog server itself more manageable.

Synchronous logging on a terminal line makes the logging process wait for the terminal to finish printing a line before outputting its own message.

Logging rate-limiting shrinks the number of messages (actually message lines) sent per second either to all destinations or to the console line specifically. Note that the RIP debugging output below only prints a limited amount of messages to the console (due to rate limiting enabled), while the logging buffer contains full messages.

```
Rack1R4#debug ip rip
RIP protocol debugging is on
RIP: received v2 update from 155.1.45.5 on Serial0/1
RIP: sending v2 update to 224.0.0.9 via Ethernet0/1 (155.1.146.4)
RIP: received v2 update from 155.1.0.5 on Serial0/0
RIP: sending v2 update to 224.0.0.9 via Serial0/0 (155.1.0.4)
RIP: received v2 update from 155.1.45.5 on Serial0/1
RIP: sending v2 update to 224.0.0.9 via Loopback0 (150.1.4.4)
RIP: sending v2 update to 224.0.0.9 via Serial0/1 (155.1.45.4)
RIP: sending v2 update to 224.0.0.9 via Ethernet0/0 (204.12.1.4)
RIP: received v2 update from 155.1.146.6 on Ethernet0/1
RIP: received v2 update from 155.1.45.5 on Serial0/1
RIP: sending v2 update to 224.0.0.9 via Loopback0 (150.1.4.4)
```



```
Rack1R4#show logging
Syslog logging: enabled (11 messages dropped, 1148 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: level debugging, 63 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level informational, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level debugging, 1177 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled

No active filter modules.

    Trap logging: level informational, 63 message lines logged

Log Buffer (8192 bytes):
0, metric 4, tag 0
       155.1.0.0/24 via 0.0.0.0, metric 1, tag 0
       155.1.5.0/24 via 0.0.0.0, metric 2, tag 0
       155.1.7.0/24 via 0.0.0.0, metric 3, tag 0
       155.1.8.0/24 via 0.0.0.0, metric 3, tag 0
       155.1.9.0/24 via 0.0.0.0, metric 4, tag 0
       155.1.10.0/24 via 0.0.0.0, metric 4, tag 0
       155.1.13.0/24 via 0.0.0.0, metric 2, tag 0
       155.1.23.0/24 via 0.0.0.0, metric 3, tag 0
       155.1.37.0/24 via 0.0.0.0, metric 3, tag 0
       155.1.45.0/24 via 0.0.0.0, metric 1, tag 0
       155.1.58.0/24 via 0.0.0.0, metric 2, tag 0
       155.1.67.0/24 via 0.0.0.0, metric 2, tag 0
       155.1.79.0/24 via 0.0.0.0, metric 3, tag 0
       155.1.108.0/24 via 0.0.0.0, metric 3, tag 0
RIP: build update entries
```


12.3 Syslog Logging

- Configure R4 and R6 to log messages to syslog as follows:
  - Both R4 and R6 should log to the server 155.X.146.100.
  - Log all messages up to notifications.
  - R4 should use reliable transport at port 5000.
  - R6 should use the default transport.
  - Use IDs ROUTER4 and ROUTER6 respectively and the UNIX facility LOCAL1.
  - Messages should be sourced off of the routers' Loopback0 interfaces.
  - Set the message queue depth to 256.

Configuration

```
R4:
logging queue-limit trap 256
logging origin-id string ROUTER4
logging facility local1
logging trap notifications
logging source-interface Loopback0
logging host 155.1.146.100 transport tcp port 5000

R6:
logging queue-limit trap 256
logging origin-id string ROUTER6
logging facility local1
logging trap notifications
logging source-interface Loopback0
logging host 155.1.146.100
```

Verification

Note

IOS supports both TCP and UDP to transport syslog messages. UDP is the default, but does not have reliable delivery like TCP does. The UNIX facility LOCAL1 allows the syslog server to multiplex incoming messages. Multiple logging destinations can be configured using their own transport, but all servers share the same facility and trap level.


```
Rack1R6#show logging
Syslog logging: enabled (11 messages dropped, 0 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: level notifications, 24 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: disabled, xml disabled,
                    filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled

No active filter modules.

    Trap logging: level notifications, 59 message lines logged
        Logging to 155.1.146.100(global) (udp port 514, audit disabled,  link
up), 2 message lines logged, xml disabled,
              filtering disabled
```

To trace the packets sent to the syslog server configure IP packet debugging for an access-list matching the syslog UDP packets. To generate a notification log go to global configuration and then type **exit**.

```
Rack1R6(config)#access-list 100 permit udp any any eq 514
Rack1R6#debug ip packet detail 100
IP packet debugging is on (detailed) for access list 100

Rack1R6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#exit
Rack1R6#
%SYS-5-CONFIG_I: Configured from console by console
IP: tableid=0, s=150.1.6.6 (local), d=155.1.146.100
(FastEthernet0/0.146), routed via RIB
```


12.4 Logging Counting and Timestamps

- Configure R4 and R6 to count error messages.
- R4 should insert uptime-based timestamps in log messages, while R6 should insert date-time based timestamps.
- R6 should add milliseconds based timestamps to debugging messages only.
- Include year information in logging messages on R6 but not in debugging messages.
- Provide a solution to prevent against tampering with stored syslog information.

Configuration

```
R4:
service timestamp debug uptime
service timestamps log uptime
service sequence-numbers
logging count

R6:
service timestamps debug datetime msec
service timestamps log datetime year
service sequence-numbers
logging count
```

Verification

Note

Timestamps can be configured separately for debugging and logging messages. They can be based on either router's uptime, or the current date and time. In the latter case, millisecond resolution can also be configured.

In the output below note that debugging message and logging output (e.g. system events) use different timestamps (logging messages have year information).


```
Rack1R6#debug ip rip
RIP protocol debugging is on
Rack1R6#
000081: Jul XX 23:25:54.920: RIP: ignored v2 update from bad source
54.1.3.254 on Serial0/0
000136: Jul XX 23:25:56.535: RIP: sending v2 update to 224.0.0.9 via
FastEthernet0/0.67 (155.1.67.6)

Rack1R6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#exit
Rack1R6#
001127: Jul XX 2008 23:32:36: %SYS-5-CONFIG_I: Configured from console
by console
```

Compare the above debugging output with the debugging output on R4, which has timestamps based on uptime.

```
Rack1R4#debug ip rip

002601: 02:55:20: RIP: sending v2 update to 224.0.0.9 via Ethernet0/1
(155.1.146.4)
002629: 02:55:22: RIP: received v2 update from 204.12.1.254 on
Ethernet0/0
002669: 02:55:23: RIP: sending v2 update to 224.0.0.9 via Loopback0
(150.1.4.4)
002736: 02:55:24: RIP: sending v2 update to 224.0.0.9 via Serial0/0
(155.1.0.4)
```

Note that all above debugging/logging lines have sequence numbers. This helps preventing tampering with the stored syslog message. Error messages counting allows to count all “notification” and above system message to provide statistics for quick analysis of the system history.


```
Rack1R6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#interface fastEthernet 0/0
Rack1R6(config-if)#shutdown
Rack1R6(config-if)#no shut
Rack1R6(config-if)#
Rack1R6(config-if)#^Z

Rack1R6#show logging count

Facility       Message Name                      Sev Occur      Last Time
================================================================================
SYS            CONFIG_I                            5    4 Jul XX 2008 23:39:00
-------------  --------------------------------  -------------------------------
SYS TOTAL                                               4

LINEPROTO      UPDOWN                              5    2 Jul XX 2008 23:38:37
-------------  --------------------------------  -------------------------------
LINEPROTO TOTAL                                         2

LINK           UPDOWN                              3    1 Jul XX 2008 23:38:36
LINK           CHANGED                             5    1 Jul XX 2008 23:38:28
-------------  --------------------------------  -------------------------------
LINK TOTAL                                              2
```

12.5 Logging to Flash Memory

- Configure SW1 to log notification messages in its internal flash memory.
- Create the directory /var/log and store messages in the file named "syslog".
- Set the maximum file size to 32768 bytes.

Configuration

```
Rack1SW1#mkdir flash:/var
Create directory filename [var]?
Created dir flash:/var

Rack1SW1#mkdir flash:/var/log
Create directory filename [/var/log]?
Created dir flash:/var/log

SW1:
logging file flash:/var/log/syslog 32768 notifications
logging on
```

Verification

> **Note**
>
> From the equipment currently available in the lab exam only the Catalyst switches support the storing of syslog messages in flash memory.

```
Rack1SW1#show logging
Syslog logging: enabled (0 messages dropped, 1 messages rate-limited, 0
flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: level notifications, 74 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level debugging, 157 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    File logging: file flash:/var/log/syslog,
        max size 32768, min size 0,
        level notifications, 2 messages logged
    Trap logging: level informational, 160 message lines logged

Log Buffer (4096 bytes):
<snip>

Rack1SW1#more flash:/var/log/syslog
03:54:30: %SYS-5-CONFIG_I: Configured from console by console
03:55:56: %SYS-5-CONFIG_I: Configured from console by console
```


12.6 Configuration Change Notification and Logging

- Configure R4 to locally track changes made to its running configuration.
- Log these changes via syslog, but ensure passwords in the configuration will not be sent across this communication channel.
- Set the queue size to 1000.

Configuration

```
R4:
archive
 log config
  logging enable
  logging size 1000
  notify syslog
  hidekeys
```

Verification

> **Note**
>
> Change notification and logging is simple alternative to AAA accounting. This feature allows you to track configuration changes along with the user who made them. The primary purpose is to log the changes to a remote syslog server for further archiving and analysis, but it's also possible to review the local changes queue.

```
Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#interface FastEthernet 0/0
004299: 02:36:49: %PARSER-5-CFGLOG_LOGGEDCMD: User:console  logged
command:interface FastEthernet0/0
Rack1R4(config-if)#shutdown
004300: 02:36:53: %PARSER-5-CFGLOG_LOGGEDCMD: User:console  logged
command:shutdown
Rack1R4(config-if)#no shutdown
004301: 02:36:56: %PARSER-5-CFGLOG_LOGGEDCMD: User:console  logged
command:no shutdown
004302: 02:36:58: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed
state to up
004303: 02:37:00: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/0, changed state to up
```


It's possible to review the changes made by all users, or just by a specified user. The statistics option displays the number of user sessions tracked and the memory usage.

```
Rack1R4#show archive log config all
 idx   sess           user@line      Logged command
    1     1        console@console  | logging enable
    2     1        console@console  |do sh run | beg arch
    3     2        console@console  |interface FastEthernet0/0
    4     2        console@console  | shutdown
    5     2        console@console  | no shutdown

Rack1R4#show archive log config statistics
Config Log Session Info:
      Number of sessions being tracked: 1
      Memory being held: 3910 bytes
      Total memory allocated for session tracking: 3910 bytes
      Total memory freed from session tracking: 0 bytes

Config Log log-queue Info:
      Number of entries in the log-queue: 5
      Memory being held by the log-queue: 1089 bytes
      Total memory allocated for log entries: 1089 bytes
      Total memory freed from log entries: 0 bytes
```

Additionally it's possible to display the changes in a format suitable to direct application to a running IOS router (e.g. to replay the changes).

```
Rack1R4#show archive log config all provisioning
logging enable
do sh run | beg arch
interface FastEthernet0/0
 shutdown
 no shutdown
```


12.7 Configuration Archive & Rollback

- Configure SW1 to store its configuration archive on the TFTP server 155.X.58.100 using “sw1-config” as the prefix.
- Archive the configuration every 24 hours.
- Additionally the configuration should be archived every time the running config is saved to NVRAM.

Configuration

```
SW1:
archive
 path tftp://155.1.58.100/sw1-config
 write-memory
 time-period 1440
```

Verification

> **Note**
>
> A router can save its configuration snapshots to a network path or to a secondary non-volatile storage plugged into router (e.g. disk0), but not to its primary flash memory storage. When saving the configuration to a network path the maximum number of copies can not be specified.

```
Rack1SW1#show archive
The next archive file will be named tftp://155.1.58.100/sw1-config -1
 Archive #  Name
   0
   1
   2
   3
   4
   5
   6
   7
   8
   9
   10
   11
   12
   13
   14
```


Configuration archiving is very useful in combination with two other features, configuration rollback and contextual diff utility. To see how they work manually create a copy of the running-configuration in flash, then make some changes to the running configuration.

```
Rack1SW1#copy running-config flash:/saved-config
Destination filename [saved-config]?

2582 bytes copied in 1.241 secs (2081 bytes/sec)

Rack1SW1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW1(config)#int vlan 67
Rack1SW1(config-if)#shutdown
Rack1SW1(config-if)#ip address 155.1.76.7 255.255.255.0
```

Now list the differences between the two configurations, the saved one and the running one. The "+" sign means that the configuration is present in the second configuration but missing in the first file. The "-" sign means the configuration line is present in the first file but is missing in the second one.

```
Rack1SW1#show archive config differences flash:/saved-config
system:/running-config
Contextual Config Diffs:
interface Vlan67
 +ip address 155.1.76.7 255.255.255.0
 +shutdown
interface Vlan67
 -ip address 155.1.67.7 255.255.255.0
```

The below output displays the lines that need to be added to the running configuration in order to make it look like the saved configuration.

```
Rack1SW1#show archive config incremental-diffs flash:/saved-config
!List of Commands:
interface Vlan67
 ip address 155.1.67.7 255.255.255.0
end
```


Use the **`configure replace`** command to roll-back the running configuration to the configuration stored in flash. This procedure will ensure the configurations are identical, and will even un-shutdown the interfaces.

```
Rack1SW1#configure replace flash:/saved-config list force
!Pass 1

!List of Commands:
interface Vlan67
 no ip address 155.1.76.7 255.255.255.0
 no shutdown
interface Vlan67
 ip address 155.1.67.7 255.255.255.0
end


Total number of passes: 1
Rollback Done
```


12.8 Logging with Access-Lists

- Configure R4 to log every RIP packet entering the interface connected to VLAN 146.
- Generate a cumulative log entry for every 2 matched packets.
- Send a logging message no more than once per 10 seconds.
- Log the MAC address of the device sending the packet.

Configuration

```
R4:
ip access-list extended LOGGING
 permit udp any any eq rip log-input
 permit ip any any
!
interface FastEthernet 0/1
 ip access-group LOGGING in
!
ip access-list logging interval 10
ip access-list log-update threshold 2
```

Verification

Note

The log-input option logs the MAC address of the device that sent the packet in addition to the packet information. The update threshold set the minimal number of hits that are needed to generate a logging message, and effectively aggregates the hits. The logging interval specifies the minimal interval of log messages generation, essentially providing a rate-limiting capability.

In the below output R1's connection to VLAN 146 is disabled to show only the updates that R4 receives from R6. Note that a log message is generated approximately every 20 seconds, as R6 is configured to send updates every 10 seconds, and the logging configuration aggregates the ACL hits in pairs.


```
Rack1R1#conf t
Rack1R1(config)#inteface FastEthernet 0/0
Rack1R1(config-if)#shutdown

Rack1R4
004318: 03:57:24: %SEC-6-IPACCESSLOGP: list LOGGING permitted udp
155.1.146.6(520) (FastEthernet0/1 0011.9283.0400) -> 224.0.0.9(520), 2
packets

004319: 03:57:43: %SEC-6-IPACCESSLOGP: list LOGGING permitted udp
155.1.146.6(520) (FastEthernet0/1 0011.9283.0400) -> 224.0.0.9(520), 2
packets

004320: 03:58:03: %SEC-6-IPACCESSLOGP: list LOGGING permitted udp
155.1.146.6(520) (FastEthernet0/1 0011.9283.0400) -> 224.0.0.9(520), 2
packets

004321: 03:58:21: %SEC-6-IPACCESSLOGP: list LOGGING permitted udp
155.1.146.6(520) (FastEthernet0/1 0011.9283.0400) -> 224.0.0.9(520), 2
packets
```

Now change the update count to 1 so that every packet hit generates a message. Note that logging messages are generated every 10 seconds now.

```
Rack1R4(config)#ip access-list log-update threshold 1

004330: 03:59:43: %SEC-6-IPACCESSLOGP: list LOGGING permitted udp
155.1.146.6(520) (FastEthernet0/1 0011.9283.0400) -> 224.0.0.9(520), 1
packet

004331: 03:59:52: %SEC-6-IPACCESSLOGP: list LOGGING permitted udp
155.1.146.6(520) (FastEthernet0/1 0011.9283.0400) -> 224.0.0.9(520), 1
packet

004332: 04:00:01: %SEC-6-IPACCESSLOGP: list LOGGING permitted udp
155.1.146.6(520) (FastEthernet0/1 0011.9283.0400) -> 224.0.0.9(520), 1
packet

004333: 04:00:11: %SEC-6-IPACCESSLOGP: list LOGGING permitted udp
155.1.146.6(520) (FastEthernet0/1 0011.9283.0400) -> 224.0.0.9(520), 1
packet
```


12.9 TCP Keepalives

- Configure R1 so that it's capable of detecting dead TCP connections and closing them prematurely.

Configuration

```
R1:
service tcp-keepalives-out
service tcp-keepalive-in
```

Verification

> **Note**
>
> The TCP keepalive feature is useful for probing idle connection to see if they are still active. In some cases, such as when the exec-timeout feature is disabled on VTY lines, this helps to prevent resources starvation by freeing connections that are not active anymore.
>
> To test this, initiate a connection from R6 to R1 and verify its parameters. Note the keepalive fields in the TCP Connection Block output.

```
Rack1R6#telnet 155.1.146.1
Trying 155.1.146.1 ... Open

User Access Verification

Password:
Rack1R1>

Rack1R1#show tcp brief all
TCB       Local Address           Foreign Address        (state)
849905B4  155.1.146.1.23          155.1.146.6.17316      ESTAB
8551BA24  *.80                    *.*                    LISTEN

Rack1R1#show tcp tcb 849905B4
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 155.1.146.1, Local port: 23
Foreign host: 155.1.146.6, Foreign port: 17316

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x1BE3ED2):
Timer          Starts    Wakeups            Next
Retrans             4          0             0x0
TimeWait            0          0             0x0
AckHold             5          2             0x0
SendWnd             0          0             0x0
KeepAlive           7          0       0x1BECF6D
```

```
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss: 3817118842  snduna: 3817118922  sndnxt: 3817118922     sndwnd:
4049
irs: 2400837743  rcvnxt: 2400837781  rcvwnd:       4091  delrcvwnd:
37

SRTT: 124 ms, RTTO: 1405 ms, RTV: 1281 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: passive open, active open, retransmission timeout,
  keepalive running
IP Precedence value : 0

Datagrams (max data segment is 1460 bytes):
Rcvd: 11 (out of order: 0), with data: 7, total data bytes: 37
Sent: 10 (retransmit: 0, fastretransmit: 0, partialack: 0, Second
Congestion: 0), with data: 7, total data bytes: 79
Rack1R1#
```

Debug TCP transactions on R1, then shut down the switch-port connected to R6 to simulate a dead connection. Note that after four keepalives have been missed R1 forcefully closes the idle connection.

```
Rack1R1#debug ip tcp transactions

Rack1SW2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1SW2(config)#int fastEthernet 0/6
Rack1SW2(config-if)#

Rack1R1#
TCP66: keepalive timeout (0/4)
Rack1R1#
TCP66: keepalive timeout (1/4)
Rack1R1#
TCP66: keepalive timeout (2/4)
Rack1R1#
TCP66: keepalive timeout (3/4)
Rack1R1#
TCP66: keepalive timeout (4/4)
TCP66: state was ESTAB -> CLOSED [23 -> 155.1.146.6(17316)]
TCB 0x849905B4 destroyed
```


12.10 Generating Exception Core Dumps

- Configure R3 to save core dumps to an FTP server at 155.X.146.100 under the name “r3-core”.
- Use active FTP and the username/password values cisco/cisco.
- Configure the router to generate a memory dump and reload as soon as free memory falls below 1Mbyte.
- The router should also reload in the case that memory fragmentation prohibits a process from allocating more than 64Kbytes of memory.
- Disable the local crash information collection.

Configuration

```
R3:
exception core-file r3-core
exception protocol ftp
exception dump 155.1.146.100
exception memory fragment 64000
exception memory minimum 1000000
!
no ip ftp passive
ip ftp username cisco
ip ftp password cisco
!
no exception crashinfo
```

Verification

> **Note**
>
> The exception feature allows for the generation of memory dumps and collecting crash information in case of emergency conditions. Memory dumps are usually stored at an outside server and delivered using FTP, TFTP, or the RCP protocol. By default, crash information is stored in local the flash memory. In this case it is explicitly disabled with the **`no exception crashinfo`** command. Special conditions may also be configured to generate exceptions and reload the router, such as when memory fragmentation occurs.
>
> When using FTP to deliver the crash dump, configure the username and password using the IOS FTP client parameters. To verify this configuration issue the **`show exception`** and **`write core`** commands.

```
Rack1R3#show exception
155.1.146.100
```


12.11 Conditional Debugging

- Configure R6 to debug RIP updates entering and leaving the connection to VLAN 67 only.

Configuration

```
Rack1R6#debug condition interface fastEthernet 0/0.67
Rack1R6#debug ip rip
```

Verification

> **Note**
>
> Conditional debugging is an extremely useful tool to limit the amount of information on busy routers. Not only is it possible to match interface names, but calling numbers (e.g. for ISDN access-servers) or usernames (e.g. for PPP sessions) are also supported. Note that the **`undebug all`** command does not remove the debug conditions. It is required to disable the conditions using explicit command syntax.
>
> Observe the debugging output on R6 with the condition set to match the VLAN 67 interface.

```
Rack1R6#
001525: Jul XX 00:00:11.026: RIP: received v2 update from 155.1.67.7 on
FastEthernet0/0.67
Rack1R6#
001531: Jul XX 00:00:15.061: RIP: sending v2 update to 224.0.0.9 via
FastEthernet0/0.67 (155.1.67.6)
Rack1R6#
001571: Jul XX 00:00:20.582: RIP: received v2 update from 155.1.67.7 on
FastEthernet0/0.67
Rack1R6#
001577: Jul XX 00:00:24.168: RIP: sending v2 update to 224.0.0.9 via
FastEthernet0/0.67 (155.1.67.6)
```

Disable the condition on R6 and see how the number of debugging messages magnifies.


```
Rack1R6#undebug condition interface fastEthernet 0/0.67
This condition is the last interface condition set.
Removing all conditions may cause a flood of debugging
messages to result, unless specific debugging flags
are first removed.

Proceed with removal? [yes/no]: yes
Condition 1 has been removed

Rack1R6#
004607: Jul 17 00:10:29.857: RIP: sending v2 update to 224.0.0.9 via
Serial0/0 (54.1.1.6)

004653: Jul 17 00:10:30.462: RIP: sending v2 update to 224.0.0.9 via
Loopback0 (150.1.6.6)

004700: Jul 17 00:10:32.445: RIP: received v2 update from 155.1.67.7 on
FastEthernet0/0.67

004706: Jul 17 00:10:33.479: RIP: sending v2 update to 224.0.0.9 via
FastEthernet0/0.146 (155.1.146.6)
```


12.12 Telnet Service Options

- Configure R3 to source all telnet sessions from its Loopback0 interface, and to use a marking of IP Precedence 3 for these packets.
- Do not display any telnet informational messages.
- Create a hostname entry “R4” that pointing to the IP address of R4.
- Ensure that a user typing “R4” in exec mode is connected without seeing the IP address of R4.
- Idle outgoing telnet sessions should signal the remote host to pause output.
- Display the message “Sorry, your connection failed” when a telnet connection to the above host fails.

Configuration

```
R3:
service telnet-zeroidle
ip telnet source-interface Loopback0
ip telnet tos 60
ip telnet quiet
ip telnet hidden addresses
!
no ip domain-lookup
ip host R4 155.1.146.4
!
busy-message R4 # Sorry, your connection failed #
```


Verification

> **Note**
>
> The telnet client in IOS can be tuned to provide additional “obfuscation” settings, which hides information from the user initiating the session. Note how the below changes once the “quiet” and “hidden address” features are disabled. With both features on, telnet never displays any progress or informational messages, and does not report the IP address of the host it connects to.

```
Rack1R3#R4
Translating "R4"

User Access Verification

Password: cisco
Rack1R4>exit

Rack1R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R3(config)#no ip telnet quiet
Rack1R3(config)#do telnet R4
Trying R4 address #1 ... Open


User Access Verification

Password: cisco
Rack1R4>exit

Rack1R3(config)#no ip telnet hidden address
Rack1R3(config)#do telnet R4
Trying R4 (155.1.146.4)... Open


User Access Verification

Password: cisco
Rack1R4>
```


To verify the TOS byte settings are correct connect to R4 from R3, then telnet back and verify the TCP connection properties. Note that the TOS value is entered in HEX format in the configuration.

```
Rack1R3#R4
Translating "R4"
Trying R4 (155.1.146.4)... Open


User Access Verification

Password: cisco
Rack1R4>telnet 150.1.3.3
Trying 150.1.3.3 ... Open


User Access Verification

Password: cisco
Rack1R3>en
Password: cisco

Rack1R3#show tcp brief all
TCB       Local Address           Foreign Address        (state)
64FEBBC0  150.1.3.3.22596         R4.23                  ESTAB
64FEC074  150.1.3.3.23            R4.52354               ESTAB
64F300E0  *.80                    *.*                    LISTEN

Rack1R3#show tcp tcb 64FEBBC0
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 150.1.3.3, Local port: 22596
Foreign host: 155.1.146.4, Foreign port: 23

<snip>

SRTT: 300 ms, RTTO: 304 ms, RTV: 4 ms, KRTT: 0 ms
minRTT: 16 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: active open, retransmission timeout
IP Precedence value : 3

Datagrams (max data segment is 536 bytes):
Rcvd: 92 (out of order: 0), with data: 86, total data bytes: 673
Sent: 95 (retransmit: 0, fastretransmit: 0, partialack: 0, Second
Congestion: 0), with data: 54, total data bytes: 125
```


To verify the busy message settings shutdown the interface connecting to VLAN 146 on R4 and then try connecting to R4 from R3. Note that the busy-message can be set globally as well as per VTY line, and is always bound to a target host-name.

```
Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#interface FastEthernet 0/1
Rack1R4(config-if)#shutdown

Rack1R3#R4
Translating "R4"
 Sorry, your connection failed
```

With the zero-idle feature an idle session will advertise a windows size of zero to the remote peer. The result of this is that all output collected at the remote side will be paused and buffered until the user resumes the session. To test this generate logging output on R6 as follows.

```
Rack1R3#telnet 150.1.6.6
Trying 150.1.6.6 ... Open

User Access Verification

Password:
Rack1R6>en
Password: cisco

Rack1R6#debug ip rip
RIP protocol debugging is on
Rack1R6#terminal monitor
Rack1R6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#logging monitor debugging
Rack1R6(config)#
005493: Jul XX 07:13:30.435: RIP: sending v2 update to 224.0.0.9 via
FastEthernet0/0.146 (155.1.146.6)
005494: Jul XX 07:13:30.435: RIP: build update entries
```

At this moment go back to R6's console and disable all debugging. Next, resume the initiated from R3 and you should see all the output collected displayed at once on the screen.

```
Rack1R6#
006485: Jul XX 07:14:17.175:     212.18.1.0/24 via 0.0.0.0, metric 2, tag 0
006486: Jul XX 07:14:17.175:     212.18.2.0/24 via 0.0.0.0, metric 2, tag 0
006487: Jul XX 07:14:17.175:     212.18.3.0/24 via 0.0.0.0, metric 2, tag 0
<snip>
```


12.13 Tuning Packet Buffers

- Enable automatic buffer tuning on R3.
- Configure R4 to double the permanent number of all public (not interface specific) buffers.
- Ensure there are at least 10 Large and Huge buffers always available.

Configuration

```
R3:
buffers tune automatic

R4:
buffers small permanent 100
buffers middle permanent 50
buffers big permanent 100
buffers verybig permanent 20
buffers large permanent 10
buffers huge permanent 10
```

Verification

Note

Buffers represent chunks of space in the router's I/O memory. Each interface may have its own private buffer pool, and all interfaces have access to public buffer pools. Buffers are used to store incoming packets, but since packets may vary in size, different group of buffer sizes exist. Having dynamically sized buffers is inefficient in terms of speed and memory fragmentation. Having pre-allocated buffers may speed-up incoming packets processing, but consumes additional memory. Therefore tuning the buffer space is a trade-off between speed and resource usage.

Sometimes it is necessary to tune buffers manually based on their historical demand. However with recent IOS versions it is possible to enable automatic buffer tuning, which changes the number of pre-allocated buffers based on an adaptive learning algorithm.

Note that for each buffer group it is possible to specify the upper and lower bounds of the buffer count, as well as the permanent and initial buffer allocation. Use the **`show buffer`** command before applying any new settings to observe the historical statistics and the current settings. A buffer "hit" mean a buffer was available for use when a packet arrived, while a "miss" means the IOS had to allocate a new buffer on demand for the packet.


```
Rack1R4#show buffers
Buffer elements:
     1118 in free list (1119 max allowed)
     79500 hits, 0 misses, 619 created

Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
     49 in free list (20 min, 150 max allowed)
     28487 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Middle buffers, 600 bytes (total 25, permanent 25, peak 31 @ 09:48:39):
     25 in free list (10 min, 150 max allowed)
     84985 hits, 2 misses, 6 trims, 6 created
     0 failures (0 no memory)
Big buffers, 1536 bytes (total 50, permanent 50):
     50 in free list (5 min, 150 max allowed)
     1595 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
VeryBig buffers, 4520 bytes (total 10, permanent 10):
     10 in free list (0 min, 100 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Large buffers, 5024 bytes (total 0, permanent 0):
     0 in free list (0 min, 10 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Huge buffers, 18024 bytes (total 0, permanent 0):
     0 in free list (0 min, 4 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
```

The below output shows the buffer sizes after they been changed on R4.

```
Rack1R4#show buffers
Buffer elements:
     1118 in free list (1119 max allowed)
     80566 hits, 0 misses, 619 created

Public buffer pools:
Small buffers, 104 bytes (total 100, permanent 100):
     99 in free list (20 min, 150 max allowed)
     29016 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
Middle buffers, 600 bytes (total 50, permanent 50):
     50 in free list (10 min, 150 max allowed)
     86380 hits, 2 misses, 6 trims, 6 created
     0 failures (0 no memory)
Big buffers, 1536 bytes (total 100, permanent 100):
     100 in free list (5 min, 150 max allowed)
     1620 hits, 0 misses, 0 trims, 0 created
     0 failures (0 no memory)
VeryBig buffers, 4520 bytes (total 20, permanent 20):
     20 in free list (0 min, 100 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
<snip>
```


12.14 Terminal Line Settings

- Configure R3 to allow only telnet connections to VTY lines 0 through 4 without using an access-list.
- When a user is telnetted into R3 and mistypes a command in exec mode, R3 should not try to open a telnet session to the command as if it was a hostname.
- Configure VTY line 0 to listen for telnet at port 3001.
- When the virtual terminal line is busy issue the output “Sorry, the line is already in use” to the connecting user.
- Exec sessions on a VTY line should timeout after 2 minutes of inactivity; additionally a user should not be able to hold the line busy for more than 5 minutes.
- Sessions initiated from a VTY line should time out in 1 minute.
- When the console line is idle the user should see the output “Welcome to IOS”
- Allow no more than 1 session to be initiated from the console line.
- The terminal length should be no more than 20 lines.
- IP netmasks should be displayed using hex numbers.
- Allow a user to lock VTY terminal lines.

Configuration

```
R3:
line vty 0 4
 session-timeout 1
 exec-timeout 2 0
 lockable
 absolute-timeout 5
 ip netmask-format hexadecimal
 refuse-message #
Sorry, the line is already in use
#
 length 20
 transport preferred none
 transport input telnet
!
line vty 0
 rotary 1
!
line console 0
 session-limit 1
 vacant-message #
Welcome to IOS
#
```


Verification

> **Note**
>
> IOS supports multiple timeout settings for terminal sessions, which can be divided as follows. Absolute – limits the maximum amount of time the user could spend on the line. Exec – limits the time an exec shell can be idle. Session – the maximum amount of time a session opened from terminal (e.g. telnet to other router) could be idle.
>
> A refuse message is displayed when someone tries to connect to a line already in use. The vacant message is displayed when the current line is idle (not in use, e.g. no exec shell started).
>
> The transport input/output option specifies which protocols can be used to connect to/from the terminal line. The preferred transport protocol is used when no session protocol is specified at exec prompt and a station name is typed. By default the preferred transport is telnet, which is why the router tries to telnet when a command is mistyped.
>
> The lock feature allows a user to lock the current terminal session and require a password to unlock it.

```
Rack1R3#show line vty 0
   Tty Typ     Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
    130 VTY              -    -      -    -    -     10       0     0/0       -

Line 130, Location: "", Type: ""
Length: 20 lines, Width: 80 columns
Baud rate (TX/RX) is 9600/9600
Status: Ready, No Exit Banner
Capabilities: Lockable
Modem state: Ready
Group codes:    0
Special Chars: Escape  Hold  Stop  Start  Disconnect  Activation
                ^^x    none   -     -       none
Timeouts:      Idle EXEC    Idle Session   Modem Answer  Session   Dispatch
               00:02:00       00:01:00                   00:05:00    not set
                             Idle Session Disconnect Warning
                               never
                             Login-sequence User Response
                              00:00:30
                             Autoselect Initial Wait
                               not set
Modem type is unknown.
Session limit is not set.
Time since activation: never
Editing is enabled.
History is enabled, history size is 20.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed input transports are telnet.
Allowed output transports are lat pad v120 lapb-ta mop telnet rlogin ssh.
Preferred transport is none.
```



```
No output characters are padded
No special data dispatching characters
```

The netmask format can be verified as below.

```
Rack1R3#telnet 150.1.3.3
Trying 150.1.3.3 ... Open


User Access Verification

Password: cisco

Rack1R3#show interface serial 1/0
Serial1/0 is up, line protocol is up
  Hardware is CD2430 in sync mode
  Internet address is 155.1.0.3 0xFFFFFF00
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
```

Rotary groups allow bundling multiple lines into a pool and to give the option to access the pool using the dedicated TCP port number 3000+N, where N is the rotary group number. Those special port numbers can also be used as "backdoors" for telnet access. Note that the refuse message is displayed when a user attempts to connect to the busy line.

```
Rack1R3#telnet 150.1.3.3 3001
Trying 150.1.3.3, 3001 ... Open


User Access Verification

Password: cisco

Rack1R3>telnet 150.1.3.3 3001
Trying 150.1.3.3, 3001 ... Open

Sorry the line is already in use

[Connection to 150.1.3.3 closed by foreign host]
```


The console session limit can be verified as follows.

```
Rack1R3#telnet 150.1.2.2
Trying 150.1.2.2 ... Open


User Access Verification

Password: cisco
Rack1R2>Ctrl-Shift-6-x
Rack1R3#telnet 150.1.5.5
Session limit exceeded

Rack1R3#w
Conn Host                Address             Byte  Idle Conn Name
*  1 150.1.2.2           150.1.2.2              0     0 150.1.2.2
```

To verify session locking login in via telnet and issuing the **lock** command.

```
Rack1R3#telnet 150.1.3.3
Trying 150.1.3.3 ... Open

User Access Verification

Password: cisco
Rack1R3>lock
Password: cisco
Again: cisco

<snip>

      Locked

Rack1R3#telnet 150.1.3.3
Trying 150.1.3.3 ... Open
```


Verify that setting the preferred transport to “none” disables automatic telnet sessions opening for hostnames entered in the CLI as follows.

```
User Access Verification

Password: cisco

Rack1R3>hostname
         ^
% Invalid input detected at '^' marker.
```

Check the vacant message for the console line as follows.

```
Rack1R3#exit

<snip>

Welcome to IOS
```

12.15 SNMPv2 Server

- Configure SNMP on R4 with the SNMP contact set to “Default Contact” and the SNMP Location to “Default Location”.
- Enable the SNMP server service using the community string value of CISCO so that remote hosts can modify the local MIB database.
- Allow the system to be reloaded via SNMP.
- Ensure that interface index numbers persist between reloads.

  Only allow configuration transfers via TFTP to/from the host 155.X.146.100.

Configuration

```
R4:
snmp-server community CISCO RW
snmp-server location Default Location
snmp-server contact Default Contact
snmp-server ifindex persist
snmp-server system-shutdown
!
access-list 98 permit 155.1.146.100
snmp-server tftp-server-list 98
```

Verification

Note

By default when you configure the SNMP agent with a community string, SNMPv2c is used. This version uses only the community string for authentication. Issuing the **`snmp-server community`** command is the minimal configuration needed to enable the devices to be polled via SNMP. The read-write string allows the management station to make changes to the managed device, as opposed to the read-only string.

SNMP interface index (IfIndex) values are not be the same between reloads by default. This may confuse some network management systems (NMS), as interfaces are identified by the IfIndex value in the SNMP Management Information Base (MIB). To resolve this the SNMP IfIndex persistence feature may be helpful.

To allow the NMS to reload the managed device, read-write access must be allowed, and the **`snmp-server system-shutdown`** feature must be enabled.

The TFTP server list specifies the acceptable addresses to download/upload the router’s configuration when instructed via SNMP.


```
Rack1R4#show snmp
Chassis: 26388555
Contact: Default Contact
Location: Default Location
0 SNMP packets input
    0 Bad SNMP version errors
    0 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    0 Number of requested variables
    0 Number of altered variables
    0 Get-request PDUs
    0 Get-next PDUs
    0 Set-request PDUs
    0 Input queue packet drops (Maximum queue size 1000)
0 SNMP packets output
    0 Too big errors (Maximum packet size 1500)
    0 No such name errors
    0 Bad values errors
    0 General errors
    0 Response PDUs
    0 Trap PDUs

SNMP logging: disabled

Rack1R4#show snmp community

Community name: ILMI
Community Index: cisco0
Community SecurityName: ILMI
storage-type: read-only  active


Community name: CISCO
Community Index: cisco3
Community SecurityName: CISCO
storage-type: nonvolatile      active

Rack1R4#show snmp mib ifmib ifindex
Ethernet0/0: Ifindex = 1
Loopback0: Ifindex = 7
Null0: Ifindex = 6
Serial0/0: Ifindex = 3
VoIP-Null0: Ifindex = 5
Ethernet0/1: Ifindex = 2
Serial0/1: Ifindex = 4
```


12.16 SNMPv2c Access Control

- Configure R4 to restrict read-write access to only hosts in VLAN 146.
- Log any other SNMP access attempts using the community string "CISCO".
- Allow any host to access R4's MIBs in read-only mode using the community string "PUBLIC".
- Read-only access should be permitted to the "cisco" MIB subtree only.

Configuration

```
R4:
access-list 99 permit 155.1.146.0 0.0.0.255
access-list 99 deny any log
!
snmp-server view ROVIEW cisco included
snmp-server community CISCO RW 99
snmp-server community PUBLIC view ROVIEW ro
```

Verification

Note

It is highly advisable to limit access to managed SNMP devices by associating an access-list with the SNMP server community. By using the log keyword it is possible to expose other hosts that attempt to access the router via SNMP.

Additionally SNMPv2 allows the restricting of access to certain MIB values using the concept of views. A view is partition of the whole MIB tree built by including or excluding various subtrees. Note that view definition may have multiple include/exclude statement, and the latter ones take preference.


```
Rack1R4#show snmp community

Community name: ILMI
Community Index: cisco0
Community SecurityName: ILMI
storage-type: read-only  active


Community name: CISCO
Community Index: cisco4
Community SecurityName: CISCO
storage-type: nonvolatile      active     access-list: 99


Community name: PUBLIC
Community Index: cisco5
Community SecurityName: PUBLIC
storage-type: nonvolatile      active

Rack1R4#show snmp view
*ilmi system - included permanent active
*ilmi atmForumUni - included permanent active
ROVIEW cisco - included nonvolatile active
v1default iso - included permanent active
v1default internet.6.3.15 - excluded permanent active
v1default internet.6.3.16 - excluded permanent active
v1default internet.6.3.18 - excluded permanent active
v1default ciscoMgmt.394 - excluded permanent active
v1default ciscoMgmt.395 - excluded permanent active
v1default ciscoMgmt.399 - excluded permanent active
v1default ciscoMgmt.400 - excluded permanent active
```


12.17 SNMP Traps and Informs

- Configure R4 to send SNMP traps to the management host 155.X.146.100 and informs to the host 155.X.146.101.
- Only send notifications about interfaces going up or down.
- Ensure no informs are being sent about Serial interfaces changing their status.
- Use the community value of “CISCO” with the above traps and informs.

Configuration

```
R4:
snmp-server enable traps snmp linkdown linkup
snmp-server host 155.1.146.101 inform version 2c CISCO
snmp-server host 155.1.146.100 CISCO
!
interface Serial0/0
 no snmp trap link-status
!
interface Serial0/1
 no snmp trap link-status
```

Verification

Note

SNMP traps are part of SNMPv1 and SNMPv2 specifications. Additionally SNMPv2 allows sending notifications as informs, which differ from traps in that they require acknowledgement from the NMS. Informs are kept in router local queue until they are acknowledged or timeout has expired. Informs make SNMP reliable even though the transport protocol is still UDP.

The traps which are to be sent can be configured either globally or on a per-host basis. In the latter case the host settings override the global settings. Some traps however can only enabled globally, such as link up and down notifications. If the single command **`snmp-server enable traps`** is entered, all traps will be sent to all configured destinations, unless they are overridden on a per-host basis. Note that *it is required* to configure both the **`snmp-server enable traps`** and the **`snmp-server host`** commands in order to send notifications to a particular host.

Informs sending is enabled by default, and the same notifications configured with the global **`snmp-server enable traps`** command are sent to NMS hosts, provided that they are configured for informs. The same host may receive informs and traps in parallel, though this configuration is uncommon.

```
Rack1R4#show snmp host
Notification host: 155.1.146.101    udp-port: 162     type: inform
user: CISCO security model: v2c

Notification host: 155.1.146.100    udp-port: 162     type: trap
user: CISCO security model: v1
```

To verify SNMP trapping, enable SNMP packet debugging and create a link down event.

```
Rack1R4#debug snmp packet
SNMP packet debugging is on
Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#interface Ethernet 0/0
Rack1R4(config-if)#shutdown
Rack1R4(config-if)#
SNMP: Inform request, reqid 17, errstat 0, erridx 0
 sysUpTime.0 = 2019043
 snmpTrapOID.0 = snmpTraps.3
 ifIndex.1 = 1
 ifDescr.1 = Ethernet0/0
 ifType.1 = 6
 lifEntry.20.1 = administratively down
Packet sent via UDP to 155.1.146.101.162
SNMP: Queuing packet to 155.1.146.100
SNMP: V1 Trap, ent snmpTraps, addr 155.1.146.4, gentrap 2, spectrap 0
 ifIndex.1 = 1
 ifDescr.1 = Ethernet0/0
 ifType.1 = 6
 lifEntry.20.1 = administratively down
Jul 18 02:24:36.590: SNMP: Packet sent via UDP to 155.1.146.100
```


12.18 CPU and Memory Thresholds

- Configure R4 to monitor CPU usage every 5 seconds using the **process cpu** command, and to generate a rising threshold event every time the CPU usage hits 50%.
- Set up the free memory low threshold to 1000Kbytes, and reserve 512Kbytes of memory for the notification process using the **memory** command.

Configuration

```
R4:
snmp-server enable traps cpu threshold
!
memory free low-watermark processor 1000
process cpu threshold type total rising 50 interval 5
memory reserve critical 512
```

Verification

Note

CPU threshold notification allows sending SNMP traps and/or informs when the CPU usage crosses defined boundaries. A trap could be generated on the CPU value crossing the rising threshold (going above) or the falling threshold (going below). Additionally the memory threshold notification feature will post a syslog message when the amount of free memory shrinks below a defined limit.

To verify CPU threshold tracking enable SNMP packet debugging and set the CPU rising threshold to a very low value. After that, execute the **show run command** to cause a CPU spike.


```
Rack1R4#debug snmp packet
Rack1r4#conf t
Rack1R4(config)#process cpu threshold type total rising 5 interval 5
Rack1R4(config)#^Z
Rack1R4#sh run
Building configuration...

Current configuration : 2493 bytes
....

%SYS-1-CPURISINGTHRESHOLD: Threshold: Total CPU
Utilization(Total/Intr): 32%/0%, Top 3 processes(Pid/Util):  3/31%,
91/0%, 2/0%

SNMP: Inform request, reqid 21, errstat 0, erridx 0
 sysUpTime.0 = 2146590
 snmpTrapOID.0 = ciscoProcessMIB.2.0.1
 cpmCPUThresholdTable.1.2.1.1 = 5
 cpmCPUTotalTable.1.10.1 = 32
 cpmCPUTotalTable.1.11.1 = 0
 ciscoProcessMIB.1.2.3.1.5.1.3 = 31
 cpmProcessTable.1.5.1.3 = 2146497
SNMP: Packet sent via UDP to 155.1.146.101.162
SNMP: Queuing packet to 155.1.146.100
SNMP: V1 Trap, ent ciscoProcessMIB.2, addr 155.1.146.4, gentrap 6,
spectrap 1
 cpmCPUThresholdTable.1.2.1.1 = 5
 cpmCPUTotalTable.1.10.1 = 32
 cpmCPUTotalTable.1.11.1 = 0
 ciscoProcessMIB.1.2.3.1.5.1.3 = 31
Rack1R4#
 cpmProcessTable.1.5.1.3 = 2146497
SNMP: Packet sent via UDP to 155.1.146.100
```

To verify free memory threshold notifications find out the current amount of free processor memory, set the threshold to a high value, and then again to a lower value.

```
Rack1R4(config)#do show memory summmary
              Head    Total(b)     Used(b)     Free(b)   Lowest(b)  Largest(b)
Processor   6420A860   59725728    14638656    45087072    44403156    43632028
      I/O    7B00000    5242880     1743784     3499096     3499096     3499068
Critical    650EDF74     481976          52      481924      481924      481924
Critical     7C9F640      42312          52       42260       42260       42260

Rack1R4(config)#memory free low-watermark processor 50000

%SYS-4-FREEMEMLOW: Free Memory has dropped below low watermark
Pool: Processor  Free: 45087396  Threshold: 51200000

Rack1R4(config)#memory free low-watermark processor 5000

%SYS-5-FREEMEMRECOVER: Free Memory has recovered above low watermark
Pool: Processor  Free: 45087660 Threshold: 5120000
```

12.19 SNMPv3

- Create two SNMP views named NORMAL and RESTRICTED on R6.
- The NORMAL view should include the branch “iso”.
- The RESTRICTED view should include the branch “ifEntry.*.*n*”, where *n* is the ifIndex of R6’s serial interface.
- Create an SNMP group named NORMAL with the read/write view assigned to NORMAL, and a security model of “priv”.
- Assign the user named NORMAL to this group, set the SHA1 password to CISCO, and the encryption key to CISCO.
- Create an SNMP group named RESTRICTED with the read view assigned to RESTRICTED, and a security model of “auth”.
- Only users in VLAN 146 should be allowed in the SNMP group RESTRICTED.
- Assign the user named RESTRICTED to this group using the security model which only requires authentication using the password CISCO.
- Create an SNMP group named TRAP with the security model “priv”.
- Assign the user named TRAP to this group using the password and encryption key CISCO.
- Enable SNMP traps for LinkUp and LinkDown events only, and send them to the destination host 155.X.146.100 using the security model “priv” and the username TRAP.

Configuration

```
R6:
access-list 99 permit 155.1.146.0 0.0.0.255
!
snmp-server ifindex persist
snmp-server view NORMAL iso included
snmp-server view RESTRICTED ifEntry.*.3 included
!
snmp-server group NORMAL v3 priv read NORMAL write NORMAL
snmp-server group RESTRICTED v3 auth read RESTRICTED access 99
snmp-server group TRAP v3 priv
!
snmp-server user NORMAL NORMAL v3 auth sha CISCO priv des56 CISCO
snmp-server user RESTRICTED RESTRICTED v3 auth sha CISCO
snmp-server user TRAP TRAP v3 auth sha CISCO priv des56 CISCO
!
snmp-server enable traps snmp linkup linkdown
snmp-server host 155.1.146.100 traps version 3 priv TRAP
```


Verification

Note

SNMPv3 extends the previous versions of SNMP by introducing a new security model that replaces the old community-based authentication system. SMNPv3 also provides for communication privacy by means of encryption. The new concepts for SNMPv3 are the user, group, and security level.

A group defines what access rights a set of users have. This access policy controls which SNMP objects (MIBs) can be accessed for reading and writing, or which SNMP objects can generate notifications to the members of a group. The policy is defined by associating a read, write, or notify view with the group. By using a notify view, a group determines the list of notifications its users can receive. The group also defines the security *model* (SNMP version) and the security *level* (authentication and/or encryption) for its users.

If a group is defined without a *read* view, all objects are available to be read (implicit permit). Contrary to that, if a *write* or *notify* view is not defined, no write access is granted, and no objects can send notifications to members of the group (implicit deny). The notify view is usually not configured manually, and is auto-generated by the **`snmp-server host`** command when users in a group are bound to a notification target host.

The security *models* are defined as SNMPv1, SNMPv2, SNMPv3, while the security *levels* are defined as noAuthNoPriv, AuthNoPriv, and AuthPriv. noAuthNoPriv, the **`noauth`** keyword in the IOS, means no authentication and no encryption. AuthNoPriv, the **`auth`** keyword in the IOS, means authentication but no encryption. AuthPriv, the **`priv`** keyword in IOS, means authentication and encryption.

SNMPv3 can implement any of the three above security levels. SNMPv1 and SNMPv2 only support noAuthNoPriv. In the case that SNMPv3 uses noAuthNoPriv, the username serves as a replacement for the community string. All users sharing a group utilize the same security model, however the specific model settings (password and encryption key) are set per-user. Note that SNMPv3 does not send passwords in clear-text, but instead uses MD5 or SHA1 hash-based authentication. For encryption, statically configured keys are used along with a single-DES (56-bit) symmetric cipher. This means that the same key should be configured on NMS for the particular user.

Note that SNMPv3 users do not appear in the running configuration for security reasons. The basic configuration of SNMPv3 can be verified as follows.

```
Rack1R6#show snmp user

User name: TRAP
Engine ID: 800000090300001192 21DA80
storage-type: nonvolatile      active
Authentication Protocol: SHA
Privacy Protocol: DES
Group-name: TRAP

User name: NORMAL
Engine ID: 800000090300001192 21DA80
storage-type: nonvolatile      active
Authentication Protocol: SHA
Privacy Protocol: DES
Group-name: NORMAL

User name: RESTRICTED
Engine ID: 800000090300001192 21DA80
storage-type: nonvolatile      active
Authentication Protocol: SHA
Privacy Protocol: None
Group-name: RESTRICTED
```

For the TRAP group the notify view is auto-generated by the **snmp-server host** command, which bound the user (TRAP) and the group it belongs to (TRAP) to the list of notifications which are to be sent to the host.

```
Rack1R6#show snmp group
groupname: ILMI                                 security model:v1
readview : *ilmi                                writeview: *ilmi
notifyview: <no notifyview specified>
row status: active

groupname: ILMI                                 security model:v2c
readview : *ilmi                                writeview: *ilmi
notifyview: <no notifyview specified>
row status: active

groupname: TRAP                                 security model:v3 noauth
readview : <no readview specified>              writeview: <no writeview
specified>
notifyview: *tv.FFFFFFFF.FFFFFFFF.FFFFFFF0F
row status: active

groupname: TRAP                                 security model:v3 priv
readview : v1default                            writeview: <no writeview
specified>
notifyview: <no notifyview specified>
row status: active
```



```
groupname: NORMAL                              security model:v3 priv
readview : NORMAL                              writeview: NORMAL
notifyview: <no notifyview specified>
row status: active

groupname: RESTRICTED                          security model:v3 auth
readview : RESTRICTED                          writeview: <no writeview
specified>
notifyview: <no notifyview specified>
row status: active      access-list: 99

Rack1R6#show snmp view
*ilmi system - included permanent active
*ilmi atmForumUni - included permanent active
NORMAL iso - included nonvolatile active
v1default iso - included permanent active
v1default internet.6.3.15 - excluded permanent active
v1default internet.6.3.16 - excluded permanent active
v1default internet.6.3.18 - excluded permanent active
v1default ciscoMgmt.394 - excluded permanent active
v1default ciscoMgmt.395 - excluded permanent active
v1default ciscoMgmt.399 - excluded permanent active
v1default ciscoMgmt.400 - excluded permanent active
RESTRICTED ifEntry.0.3 FF:EF included nonvolatile active
*tv.FFFFFFFF.FFFFFFFF.FFFFFFFF0F iso.2.840.10036 - included volatile
active
*tv.FFFFFFFF.FFFFFFFF.FFFFFFFF0F internet - included volatile active
```

To ensure that SNMP traps are being sent in encrypted and authenticated, enable SNMP packet debugging along with a low-level packet dump for outgoing SNMP traps as follows.

```
Rack1R6(config)#access-list 100 permit udp any any eq 162
Rack1R6#debug ip packet detail 100 dump
IP packet debugging is on (detailed) (dump) for access list 100

Rack1R6#debug snmp packet
SNMP packet debugging is on

Rack1R6#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R6(config)#interface loopback 0
Rack1R6(config-if)#shutdown
Rack1R6(config-if)#

SNMP Queuing packet to 155.1.146.100:
SNMP: V2 Trap, reqid 23, errstat 0, erridx 0
 sysUpTime.0 = 2893642
 snmpTrapOID.0 = snmpTraps.3
 ifIndex.11 = 11
 ifDescr.11 = Loopback0
 ifType.11 = 24
 lifEntry.20.11 = administratively down
```



```
SNMP: Packet sent via UDP to 155.1.146.100
IP: tableid=0, s=155.1.146.6 (local), d=155.1.146.100
(FastEthernet0/0.146), routed via RIB
IP: s=155.1.146.6 (local), d=155.1.146.100 (FastEthernet0/0.146), len
283, sending
    UDP src=58135, dst=162
07802010:                   4500011B 00060000          E.......
07802020: FF11605E 9B019206 9B019264 E31700A2  ..`^.......dc.."
07802030: 01077D80 3081FC02 0103300D 02011202  ..}.0.|...0.....
07802040: 0205DC04 01030201 03043530 33040C80  ..\.......503...
07802050: 00000903 00001192 21DA8002 01180202  ........!Z......
07802060: 190C0404 54524150 040C3972 C05E1654  ....TRAP..9r@^.T
07802070: 7D2249D0 11F10408 00000018 F1F81DCD  }"IP.q......qx.M
07802080: 0481B021 3372C60A 394B07E8 93FD35A3  ..0!3rF.9K.h.}5#
07802090: 584291FA 887D96B8 6894CCE6 58AAD5DD  XB.z.}.8h.LfX*U]
078020A0: 5F4EACE4 4B30FB5C 25B58A78 09A78EF0  _N,dK0{\%5.x.'.p
078020B0: 863CEE49 3745902A FEE6530C BAD247DC  .<nI7E.*~fS.:RG\
078020C0: DF94530F E5ED993C FC955C8C 3B42FC15  _.S.em.<|.\.;B|.
078020D0: D4C2B05B 8CF2869A FED2EC8E 38570FBB  TB0[.r..~Rl.8W.;
078020E0: C1454016 599591C2 D3FA07B1 0D048B26  AE@.Y..BSz.1...&
078020F0: EF4CBD1F 34B5E63A 0C05AA56 385B32D2  oL=.45f:..*V8[2R
07802100: E5AB5AB6 F9F722F2 13C5610E 4ADE3FC3  e+Z6yw"r.Ea.J^?C
07802110: 73F78791 74AED319 5F86D648 74A04BE9  sw..t.S._.VHt Ki
07802120: 1CCDAB2E 0D022589 6BFBE813 E613B58C  .M+...%.k{h.f.5.
07802130: 739C66                               s.f
IP: s=155.1.146.6 (local), d=155.1.146.100 (FastEthernet0/0.146), len
283, encapsulation failed
    UDP src=58135, dst=162
07802010:                   4500011B 00060000          E.......
07802020: FF11605E 9B019206 9B019264 E31700A2  ..`^.......dc.."
07802030: 01077D80 3081FC02 0103300D 02011202  ..}.0.|...0.....
07802040: 0205DC04 01030201 03043530 33040C80  ..\.......503...
07802050: 00000903 00001192 21DA8002 01180202  ........!Z......
07802060: 190C0404 54524150 040C3972 C05E1654  ....TRAP..9r@^.T
07802070: 7D2249D0 11F10408 00000018 F1F81DCD  }"IP.q......qx.M
07802080: 0481B021 3372C60A 394B07E8 93FD35A3  ..0!3rF.9K.h.}5#
07802090: 584291FA 887D96B8 6894CCE6 58AAD5DD  XB.z.}.8h.LfX*U]
078020A0: 5F4EACE4 4B30FB5C 25B58A78 09A78EF0  _N,dK0{\%5.x.'.p
078020B0: 863CEE49 3745902A FEE6530C BAD247DC  .<nI7E.*~fS.:RG\
078020C0: DF94530F E5ED993C FC955C8C 3B42FC15  _.S.em.<|.\.;B|.
078020D0: D4C2B05B 8CF2869A FED2EC8E 38570FBB  TB0[.r..~Rl.8W.;
078020E0: C1454016 599591C2 D3FA07B1 0D048B26  AE@.Y..BSz.1...&
078020F0: EF4CBD1F 34B5E63A 0C05AA56 385B32D2  oL=.45f:..*V8[2R
07802100: E5AB5AB6 F9F722F2 13C5610E 4ADE3FC3  e+Z6yw"r.Ea.J^?C
07802110: 73F78791 74AED319 5F86D648 74A04BE9  sw..t.S._.VHt Ki
07802120: 1CCDAB2E 0D022589 6BFBE813 E613B58C  .M+...%.k{h.f.5.
07802130: 739C66                               s.f             -if)#
```


Change the security model for the destination host to “noAuth” and generate a trap message again. Note that now the message is not encrypted.

```
Rack1R6# snmp-server host 155.1.146.100 version 3 noauth TRAP

Rack1R6(config)#interface Loopback 0
Rack1R6(config-if)#no shut
Rack1R6(config-if)#
SNMP: Queuing packet to 155.1.146.100
SNMP: V2 Trap, reqid 24, errstat 0, erridx 0
 sysUpTime.0 = 2928473
 snmpTrapOID.0 = snmpTraps.4
 ifIndex.11 = 11
 ifDescr.11 = Loopback0
 ifType.11 = 24
 lifEntry.20.11 = up
SNMP: Packet sent via UDP to 155.1.146.100
IP: tableid=0, s=155.1.146.6 (local), d=155.1.146.100
(FastEthernet0/0.146), routed via RIB
IP: s=155.1.146.6 (local), d=155.1.146.100 (FastEthernet0/0.146), len
238, sending
    UDP src=58135, dst=162
079D76B0:                   450000EE 00070000          E..n....
079D76C0: FF11608A 9B019206 9B019264 E31700A2  ..`........dc.."
079D76D0: 00DA3F33 3081CF02 0103300D 02011302  .Z?30.O...0.....
079D76E0: 0205DC04 01000201 03042130 1F040C80  ..\.......!0....
079D76F0: 00000903 00001192 21DA8002 01180202  ........!Z......
079D7700: 1A680404 54524150 04000400 30819704  .h..TRAP....0...
079D7710: 0C800000 09030000 119221DA 800400A7  ..........!Z...'
079D7720: 81840201 18020100 02010030 79300F06  ...........0y0..
079D7730: 082B0601 02010103 0043032C AF593017  .+.......C.,/Y0.
079D7740: 060A2B06 01060301 01040100 06092B06  ..+...........+.
079D7750: 01060301 01050430 0F060A2B 06010201  .......0...+....
079D7760: 02020101 0B02010B 3017060A 2B060102  ........0...+...
079D7770: 01020201 020B0409 4C6F6F70 6261636B  ........Loopback
079D7780: 30300F06 0A2B0601 02010202 01030B02  00...+..........
079D7790: 01183012 060C2B06 01040109 02020101  ..0...+.........
079D77A0: 140B0402 7570                        ....up
<snip>
```


12.20 SNMP MAC Address Notifications

- Configure SW1 to notify the NMS station at the IP address 155.X.146.100 about MAC addresses learned on ports connected to R1 and R5.
- Use the community value of CISCO to send notifications.
- Limit the rate of notifications to 1 per second.
- Report both added and removed MAC addresses.
- Store the latest 100 notification events in history buffer.

Configuration

```
SW1:
interface FastEthernet0/1
 snmp trap mac-notification added
 snmp trap mac-notification removed
!
interface FastEthernet0/5
 snmp trap mac-notification added
 snmp trap mac-notification removed
!
snmp-server enable traps mac-notification
!
mac-address-table notification interval 2
mac-address-table notification history-size 100
mac-address-table notification
```


Verification

Note

MAC Address table notifications allow the switches to report changes in the CAM table entries (adds and removes) for a particular link. This feature must be first enabled globally, and then configured on per-interface level. On trunk interfaces all VLANs are affected by the configuration. In addition to reporting changes to an NMS the switch can store recent events in the local buffer.

```
Rack1SW1#clear mac-address-table dynamic
Rack1SW1#show mac-address-table notification
MAC Notification Feature is Enabled on the switch
Interval between Notification Traps : 2 secs
Number of MAC Addresses Added : 7
Number of MAC Addresses Removed : 7
Number of Notifications sent to NMS : 9
Maximum Number of entries configured in History Table : 100
Current History Table Length : 2
MAC Notification Traps are Enabled
History Table contents
----------------------
History Index 0, Entry Timestamp 3441456, Despatch Timestamp 3441456
MAC Changed Message :
Operation: Deleted Vlan: 58    MAC Addr: 0004.9a0b.62c1 Dot1dBasePort: 7
Operation: Deleted Vlan: 146   MAC Addr: 0011.2030.f900 Dot1dBasePort: 3
Operation: Added   Vlan: 58    MAC Addr: 0004.9a0b.62c1 Dot1dBasePort: 7
History Index 1, Entry Timestamp 3441657, Despatch Timestamp 3441657
MAC Changed Message :
Operation: Added   Vlan: 146   MAC Addr: 0011.2030.f900 Dot1dBasePort: 3
```


12.21 SNMP Notifications of Syslog Messages

- Configure R1 so that all debugging and higher priority level messages are sent via SNMP to an NMS host at the IP address 155.X.146.100.
- Set the syslog to SNMP buffer size to 100 messages.

Configuration

```
R1:
snmp-server enable traps syslog
snmp-server host 155.1.146.100 CISCO
!
logging history debugging
logging history size 100
```

Verification

> **Note**
>
> SNMP logging of syslog messages allows the router to forward syslog messages to a remote NMS station using SNMP trap PDUs. Syslog fist sends the logs to a special history buffer, then the SNMP agent replicates the messages as SNMP traps. This requires that "syslog" traps have been enabled globally or for that particular NMS.
>
> To verify this configuration enable SNMP packet debugging and generate a syslog message by shutting down an interface.

```
Rack1R1#debug snmp packets
SNMP packet debugging is on
Rack1R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R1(config)#interface serial 0/0
Rack1R1(config-if)#shutdown
Rack1R1(config-if)#^Z
Rack1R1#
SNMP: Queuing packet to 155.1.146.100
SNMP: V1 Trap, ent ciscoSyslogMIB.2, addr 155.1.146.1, gentrap 6,
spectrap 1
 clogHistoryEntry.2.18 = LINK
 clogHistoryEntry.3.18 = 6
 clogHistoryEntry.4.18 = CHANGED
 clogHistoryEntry.5.18 = Interface Serial0/0, changed state to
administratively down
 clogHistoryEntry.6.18 = 3559736
SNMP: Packet sent via UDP to 155.1.146.100
SNMP: Queuing packet to 155.1.146.100
SNMP: V1 Trap, ent ciscoSyslogMIB.2, addr 155.1.146.1, gentrap 6,
spectrap 1
 clogHistoryEntry.2.19 = SYS
```



```
 clogHistoryEntry.3.19 = 6
 clogHistoryEntry.4.19 = CONFIG_I
 clogHistoryEntry.5.19 = Configured from console by console
 clogHistoryEntry.6.19 = 3559777
SNMP: Packet sent via UDP to 155.1.146.100
SNMP: Queuing packet to 155.1.146.100
SNMP: V1 Trap, ent ciscoSyslogMIB.2, addr 155.1.146.1, gentrap 6,
spectrap 1
 clogHistoryEntry.2.20 = LINEPROTO
 clogHistoryEntry.3.20 = 6
 clogHistoryEntry.4.20 = UPDOWN
 clogHistoryEntry.5.20 = Line protocol on Interface Serial0/0, changed
state to down
 clogHistoryEntry.6.20 = 3559836
SNMP: Packet sent via UDP to 155.1.146.100
```

```
Rack1R1#show snmp
Chassis: JAE0817F53F
<snip>
3 SNMP packets output
    0 Too big errors (Maximum packet size 1500)
    0 No such name errors
    0 Bad values errors
    0 General errors
    0 Response PDUs
    3 Trap PDUs

SNMP logging: enabled
    Logging to 155.1.146.100.162, 0/10, 6 sent, 0 dropped.
```

```
Rack1R1#show logging history
Syslog History Table:100 maximum table entries,
saving level debugging or higher
 47 messages ignored, 11 dropped, 0 recursion drops
 11 table entries flushed
 SNMP notifications enabled, 6 notifications sent
<snip>
   entry number 18 : LINK-5-CHANGED
    Interface Serial0/0, changed state to administratively down
    timestamp: 3559736
   entry number 19 : SYS-5-CONFIG_I
    Configured from console by console
    timestamp: 3559777
   entry number 20 : LINEPROTO-5-UPDOWN
    Line protocol on Interface Serial0/0, changed state to down
    timestamp: 3559836
```


12.22 CDP

- Configure R4 to send CDP announcement every 10 seconds, and instruct other devices to hold the updates for 40 seconds.
- Use the Loopback0 interface for IP address advertisements in CDP messages.
- Disable logging of duplex mismatch detected via CDP messages.
- Do not send or receive CDP message on R4's connection to VLAN 43.

Configuration

```
R4:
no cdp log mismatch duplex
cdp source-interface Loopback0
cdp timer 10
cdp holdtime 40
!
interface FastEthernet 0/0
 no cdp enable
```

Verification

> **Note**
>
> Cisco Discovery Protocol (CDP) is used to exchange basic device information and aid in troubleshooting. CDP can be enabled/disabled globally with the **`[no] cdp run`** command, and at the interface level with the **`[no] cdp enable`**.

```
Rack1R4#show cdp
Global CDP information:
      Sending CDP packets every 10 seconds
      Sending a holdtime value of 40 seconds
      Sending CDPv2 advertisements is  enabled
      Source interface is Loopback0

Rack1R4#show cdp interface
FastEthernet0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 10 seconds
  Holdtime is 40 seconds
Serial0/1 is up, line protocol is up
  Encapsulation HDLC
  Sending CDP packets every 10 seconds
  Holdtime is 40 seconds
```



```
Rack1R4#show cdp traffic
CDP counters :
      Total packets output: 160, Input: 148
      Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
      No memory: 0, Invalid packet: 0, Fragmented: 0
      CDP version 1 advertisements output: 0, Input: 0
      CDP version 2 advertisements output: 160, Input: 148

Rack1SW4#show cdp neighbors fastEthernet 0/4 detail
-------------------------
Device ID: Rack1R4
Entry address(es):
  IP address: 150.1.4.4
Platform: Cisco 2611XM,  Capabilities: Router Switch IGMP
Interface: FastEthernet0/4,  Port ID (outgoing port): FastEthernet0/1
Holdtime : 33 sec

Version :
Cisco IOS Software, C2600 Software (C2600-ADVENTERPRISEK9-M), Version
12.4(10), RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Wed 16-Aug-06 01:42 by prod_rel_team

advertisement version: 2
VTP Management Domain: ''
Duplex: full
Management address(es):
```


12.23 RMON Alarms

- Configure R1 to monitor the rate of packets entering its interface connecting to VLAN 146 based on the total input packet counter (ifEntry.11).
- If the rate is above 100 packets per minute generate the log message "VLAN146 Interface Congested" and send a trap to the management host at 155.X.146.100.
- Use the SNMP community string CISCO.
- When the packet rate falls below 50 packets per minute, generate another log message "VLAN146 Interface UnCongested" and send a corresponding trap.
- The owner of events should be set to "CISCO".

Configuration

```
R1:
snmp-server host 155.1.146.100 CISCO
snmp-server ifindex persist
!
rmon event 1 log trap CISCO description "VLAN146 Interface Congested"
owner CISCO
rmon event 2 log trap CISCO description "VLAN146 Interface UnCongested"
owner CISCO
!
rmon alarm 1 ifEntry.11.1 60 delta rising-threshold 100 1 falling-
threshold 50 2 owner CISCO
```

Verification

> **Note**
>
> The Remote Monitoring (RMON) feature is used to monitor specified SNMP MIB variables and generate events (syslog messages or SNMP traps) when a threshold or change threshold is crossed. The common misconception about the RMON feature is the difference between an *absolute* sampling and a *delta* sampling.

An absolute sampling threshold is used for variables that increase or decrease over time, and have an upper or lower bound for when a log should be generated. Examples of absolute samplings would be CPU utilization, memory utilization, interface queue depth, etc. A delta sampling threshold is used for variables that either constantly increase (most common) or constantly decrease. Examples of delta samplings would be for interface errors, input packets, output bytes, etc.

For example if the CPU utilization is at 60% at interval 1, and increases to 90% at interval 2, it makes more sense to want to know that the CPU utilization is exactly 90% (absolute) versus it having changed 30% (delta) over the interval. On the other hand if we find that at interval 1 the total number of bytes received inbound on an interface is 1000, and at interval 2 the total bytes increases to 10000, it makes more sense to want to know that the inbound link utilization was 9000 bytes per interval (delta) versus having increased to 10000 bytes (absolute). The problem with the absolute sampling in this second case is that there's no way to tell if those 10000 bytes were received in the last minute or the last month.

RMON configuration consists of defining events and creating an alarm on a known MIB variable. In the presented case, the variable given is a an interface counter (lifEntry.11) and therefore to get rate from it the delta sampling method is applied. Note that the variable name is based on interface index, so additional steps are needed to discover the interface index and ensure that the indexes are persistent across reboots.

To verify the configuration enable SNMP packet debugging and generate traffic across VLAN 146 to make sure the counter increases by more than 100 packets.

```
Rack1R1#debug snmp packet
Rack1R1#ping 155.1.146.6 repeat 150
Type escape sequence to abort.
Sending 150, 100-byte ICMP Echos to 155.1.146.6, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
<snip>
Success rate is 100 percent (150/150), round-trip min/avg/max = 1/2/4 ms

%RMON-5-RISINGTRAP: Rising trap is generated because the value of
ifInUcastPkts.1 exceeded the rising-threshold value 100
Rack1R1#
SNMP: Queuing packet to 155.1.146.100
SNMP: V1 Trap, ent rmon, addr 155.1.146.1, gentrap 6, spectrap 1
 alarmEntry.1.1 = 1
 alarmEntry.3.1 = ifInUcastPkts.1
 alarmEntry.4.1 = 2
 alarmEntry.5.1 = 150
 alarmEntry.7.1 = 100
SNMP: Packet sent via UDP to 155.1.146.100
```


Do not send any packets after that, and wait for a minute.

```
%RMON-5-FALLINGTRAP: Falling trap is generated because the value of
ifInUcastPkts.1 has fallen below the falling-threshold value 50
Rack1R1#
SNMP: Queuing packet to 155.1.146.100
SNMP: V1 Trap, ent rmon, addr 155.1.146.1, gentrap 6, spectrap 2
 alarmEntry.1.1 = 1
 alarmEntry.3.1 = ifInUcastPkts.1
 alarmEntry.4.1 = 2
 alarmEntry.5.1 = 0
 alarmEntry.8.1 = 50
SNMP: Packet sent via UDP to 155.1.146.100
```

Display the RMON event and alarms configuration information and statistics as follows.

```
Rack1R1#show rmon events
Event 1 is active, owned by CISCO
 Description is VLAN146 Interface Congested
 Event firing causes log and trap to community CISCO,
 last event fired at  0y0w0d,09:39:11,
 Current uptime       0y0w0d,09:51:22
 Current log entries:
  index  uptime              description
  1      0y0w0d,09:39:11     VLAN146 Interface Congested
Event 2 is active, owned by CISCO
 Description is VLAN146 Interface UnCongested
 Event firing causes log and trap to community CISCO,
 last event fired at  0y0w0d,09:40:12,
 Current uptime       0y0w0d,09:51:22
 Current log entries:
  index  uptime              description
  1      0y0w0d,03:59:10     VLAN146 Interface UnCongested
  2      0y0w0d,09:38:11     VLAN146 Interface UnCongested
  3      0y0w0d,09:40:12     VLAN146 Interface UnCongested

Rack1R1#show rmon alarm
Alarm 1 is active, owned by CISCO
 Monitors ifInUcastPkts.1 every 60 second(s)
 Taking delta samples, last value was 0
 Rising threshold is 100, assigned to event 1
 Falling threshold is 50, assigned to event 2
 On startup enable rising or falling alarm
```


12.24 RMON Statistics Collection

- Configure SW1 to collect RMON statistics on packets entering the interface connected to R3.
- Store historical information for the last 100 samples of packet statistics taken every 30 seconds.

Configuration

```
SW1:
interface FastEthernet 0/3
 rmon promiscuous
 rmon collection history 1 owner monitor buckets 100 interval 30
```

Verification

Note

The RMON agent functionality contains a comprehensive set of statistical analysis tools. However, only a few Cisco routers and IOS versions implement the full RMON specification (notably Cisco 2500 models).

Catalyst switches have additional RMON packet statistics collection functionality which allows them to collect and store information on packets entering an interface. RMON collection can be configured using native mode (only analyze packets destined to interface) or promiscuous mode (capture all packets on the segment, even those not destined to interface).

RMON statistics collection can be configured on either layer 2 switchports or native layer 3 routed ports, but not SVIs. History collection allows accumulating packet statistics in time-intervals (buckets) and storing the buckets in the history buffer.

Statistics collection can be verified as follows.

```
Rack1SW1#show rmon statistics
Collection 10003 on FastEthernet0/3 is active, and owned by config,
 Monitors ifIndex.10003 which has
 Received 89005 octets, 397 packets,
 0 broadcast and 221 multicast packets,
 0 undersized and 0 oversized packets,
 0 fragments and 0 jabbers,
 0 CRC alignment errors and 0 collisions.
 # of dropped packet events (due to lack of resources): 0
 # of packets received of length (in octets):
  64: 176, 65-127: 1, 128-255: 0,
  256-511: 220, 512-1023: 0, 1024-1518:0
```



```
Rack1SW1#ping 155.1.37.3 size 500 repeat 100

Type escape sequence to abort.
Sending 100, 500-byte ICMP Echos to 155.1.37.3, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/2/9 ms

Rack1SW1#show rmon statistics
Collection 10003 on FastEthernet0/3 is active, and owned by config,
 Monitors ifIndex.10003 which has
 Received 194961 octets, 607 packets,
 0 broadcast and 227 multicast packets,
 0 undersized and 0 oversized packets,
 0 fragments and 0 jabbers,
 0 CRC alignment errors and 0 collisions.
 # of dropped packet events (due to lack of resources): 0
 # of packets received of length (in octets):
  64: 180, 65-127: 1, 128-255: 0,
  256-511: 226, 512-1023: 200, 1024-1518:0
```

Note that the history collection clearly displays the packet "burst" generated by the ping command.

```
Rack1SW1#show rmon history
Entry 1 is active, and owned by monitor
 Monitors ifIndex.10003 every 30 second(s)
 Requested # of time intervals, ie buckets, is 100,
  Sample # 1 began measuring at 07:06:00
   Received 2914 octets, 13 packets,
   0 broadcast and 7 multicast packets,
   0 undersized and 0 oversized packets,
   0 fragments and 0 jabbers,
   0 CRC alignment errors and 0 collisions.
   # of dropped packet events is 0
   Network utilization is estimated at 0
  Sample # 2 began measuring at 07:06:30
   Received 3228 octets, 14 packets,
   0 broadcast and 8 multicast packets,
   0 undersized and 0 oversized packets,
   0 fragments and 0 jabbers,
   0 CRC alignment errors and 0 collisions.
   # of dropped packet events is 0
   Network utilization is estimated at 0
<snip>
Sample # 20 began measuring at 07:15:30
   Received 106828 octets, 214 packets,
   0 broadcast and 8 multicast packets,
   0 undersized and 0 oversized packets,
   0 fragments and 0 jabbers,
   0 CRC alignment errors and 0 collisions.
   # of dropped packet events is 0
   Network utilization is estimated at 2
```


12.25 HTTP Server and Client

- Configure R4 to support the transfer of the IOS image stored in R6's flash using HTTP.
- R4 should authenticate with a username/password pair of CISCO/CISCO stored in the local database of R6
- R6 should only allow HTTP connections at port 8080, from the subnet 150.X.0.0/16, and limit the maximum number of concurrent connections to two.
- Change the default WWW port to 8080.
- Allow R4 to use secure HTTP connection on port 4043, but limit the security features to use only the DES symmetric cipher.

Configuration

```
R6:
username CISCO password 0 CISCO
!
ip http server
ip http max-connections 2
ip http path flash:
ip http port 8080
!
access-list 80 permit 150.1.0.0 0.0.255.255
!
ip http access-class 80
ip http authentication local
ip http secure-server
ip http secure-port 4043
ip http secure-ciphersuite des-cbc-sha

R4:
ip http client source-interface Loopback0
ip http client username CISCO
ip http client password CISCO
ip http client secure-ciphersuite des-cbc-sha
```


Verification

Note

IOS HTTP server is enabled by default to allow for remote router management (SDM). The default authentication is via the enable password (any name and the enable password) and can be changed to use the local database if AAA is not configured. Note that the local user needs privilege level 15 to access the router via HTTP. Additional security settings include setting the access-class or changing the default port number.

Secure HTTP server enables the use of SSL to protect communications. Note that by default the secure server is disabled, and automatically generates a server SSL X.509 certificate when enabled (this will also bring the SSH server up). When changing the server cipher-suite make sure the client will accept it during the handshake.

In addition to server functionality IOS implements an HTTP client. Using this client it's possible to transfer files to the local router from the remote server. The client is even capable of using a proxy server.

To verify the client/server issue a copy http command from R4.

```
Rack1R4#copy http://155.1.146.6:8080/c2600-adventerprisek9-mz.124-10.bin null:
Loading http://155.1.146.6:8080/c2600-adventerprisek9-mz.124-10.bin
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Try to connect to the HTTP server from a different source interface or using different credentials.

```
Rack1R4#conf t
Rack1R4(config)#ip http client source FastEthernet 0/0
Rack1R4(config)#^Z
Rack1R4#copy http://155.1.146.6:8080/c2600-adventerprisek9-mz.124-
10.bin null:
%Error opening http://155.1.146.6:8080/c2600-adventerprisek9-mz.124-
10.bin (I/O error)

Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#ip http client source-interface Loopback 0
Rack1R4(config)#ip http client username USER
Rack1R4(config)#exit
Rack1R4#copy http://155.1.146.6:8080/c2600-adventerprisek9-mz.124-
10.bin null:
%Error opening http://155.1.146.6:8080/c2600-adventerprisek9-mz.124-
10.bin (Permission denied)
```


Verify the HTTP settings on the server. Note that “HTTP secure server client authentication: Disabled” means that the server will not ask for the client’s SSL certificate.

```
Rack1R6#show ip http server status
HTTP server status: Enabled
HTTP server port: 8080
HTTP server authentication method: local
HTTP server access class: 0
HTTP server base path: flash:
Maximum number of concurrent server connections allowed: 2
Server idle time-out: 180 seconds
Server life time-out: 180 seconds
Maximum number of requests allowed on a connection: 1
HTTP server active session modules: ALL
HTTP secure server capability: Present
HTTP secure server status: Enabled
HTTP secure server port: 4043
HTTP secure server ciphersuite: des-cbc-sha
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint:
HTTP secure server active session modules: ALL
```


Verify the HTTP client settings on R4 as follows.

```
Rack1R4#show ip http client all

HTTP client status: Enabled
HTTP client application session modules:
Id               : 1
Application Name : HTTP CFS
Version          : HTTP/1.0
Persistent       : persistent
Response-timeout : 0
Retries          : 0
Proxy            :

HTTP client current connections:
Persistent connection = enabled (default)
Connection establishment timeout = 10s (default)
Connection idle timeout = 30s (default)
Maximum number of connection establishment retries = 1 (default)
Maximum http client connections per host : 2
HTTP secure client capability: Present
HTTP secure client ciphersuite: des-cbc-sha
HTTP secure client trustpoint:

local-ipaddress:port  remote-ipaddress:port in-bytes   out-bytes


Total client connections : 0


HTTP client cache:
Maximum Memory size for cache    : 100000 bytes (default)
Maximum memory per cache entry   : 2000 bytes (default)
Memory used                      : 0 bytes
Memory Available                 : 100000 bytes
Cache Ager interval              : 5 minutes (default)
Total entries created            : 0
Id    Type    Url              Memory-size(Bytes)     Refcnt     Valid(Sec)
_____________________________________________________________________________


HTTP client statistics:

HTTP client history:
     GET 01:10:45 UTC Wed MMM DD YYYY 150.1.6.6/c2600-adventerprisek9-mz.124...
     GET 01:12:37 UTC Wed MMM DD YYYY 150.1.6.6/c2600-adventerprisek9-mz.124...
```


12.26 FTP Server and Client

- Configure the FTP server service on R6 to serve IOS images off of flash.
- R4 should initiate an FTP session to R6 with the username/password CISCO/CISCO.
- When a transfer occurs R6 should initiate the FTP data channel back to the client.
- Source FTP connections off R4 Loopback0 interface

Configuration

```
R6:
ftp-server enable
ftp-server topdir flash:

R4:
no ip ftp passive
ip ftp source-interface Loopback0
ip ftp username CISCO
ip ftp password CISCO
```

Verification

Note

In addition to FTP client support, IOS may act as a simple FTP server. The server service does not support any authentication, and due to numerous vulnerabilities (such as bug ID CSCse29244, *IOS crash when transferring files via FTP*) Cisco has *removed* the FTP server functionality from recent IOS releases, promising to re-implement a more stable and advanced feature set at a later time. The FTP client feature can be used to save core dumps and transfer files from external FTP servers. For file transfers between IOS routers the HTTP or TFTP protocols should be used. Due to the FTP server crashing IOS, it is impossible to properly verify the FTP server functionality.

12.27 TFTP Server and Client

- Configure the TFTP server service on R6.
- When someone requests the file “r6-ios” from R6 it should return its IOS image file.
- Ensure only R1’s Loopback0 interface is allowed to connect and retrieve files via TFTP from R6.
- Configure R1 to source TFTP packets off its Loopback0 interface.

Configuration

```
R6:
tftp-server flash:c2600-adventerprisek9-mz.124-10.bin alias r6-ios 10
!
access-list 10 permit 150.1.1.1

R1:
ip tftp source-interface Loopback0
```

Verification

> **Note**
>
> IOS TFTP server allows read-only access to specified files. An access-list can be used to limit the scope of users accessing the files. The alias feature provides abbreviated names for long filenames stored in the flash.

```
Rack1R6#debug tftp events
TFTP Event debugging is on

Rack1R1#copy tftp://150.1.6.6/r6-ios null:
Accessing tftp://150.1.6.6/r6-ios...
Loading r6-ios from 150.1.6.6 (via FastEthernet0/0): !!

Rack1R6#
Looking for r6-ios
TFTP: Opened flash:c2600-adventerprisek9-mz.124-10.bin, fd 0, size
29631128 for process 68
TFTP: Finished flash:c2600-adventerprisek9-mz.124-10.bin, time 00:00:00
for process 68
```

R4 cannot access the same file using TFTP.

```
Rack1R4#copy tftp://150.1.6.6/r6-ios null:
Accessing tftp://150.1.6.6/r6-ios...

Rack1R6#
TFTP: Looking for r6-ios
TFTP: Looking for r6-ios
```


12.28 Remote Shell

- Configure the network so that R1 is to view the running configuration of R6 using remote shell.
- Source remote-shell connections off the Loopback0 interface of R1
- Allow RCP file-copy from R6 to R1.

Configuration

```
R1:
ip rcmd remote-username RCP
ip rcmd source-interface Loopback0

R6:
ip rcmd rcp-enable
ip rcmd rsh-enable
!
ip rcmd remote-host Rack1R6 150.1.1.1 Rack1R1 enable
ip rcmd remote-host RCP 150.1.1.1 Rack1R1 enable
```

Verification

Note

IOS implementation includes the UNIX-line remote command execution environment (RSH). No password is required to access the local system, just an entry in local ".rhosts"-like table. The table maps a remote station's IP and username to the local username and privilege level (in a classic UNIX system, the privilege level is determined by the local username) using the following syntax:

ip rcmp remote-host <local-name> <remote-IP> <remote-name> <privilege>

The local name in the ".rhosts" table does not need to match any user in the loca database, and commonly the target router's hostname is used as the RCMD local username. When a remote host connects to the local router via remote-shell the remote host's IP address is looked up in ".rhosts" table, the target username specified by remote system is matched against <local-name>, and the source username specified by remote system is matched against <remote-name>. If all fields are matched, remote-shell access is granted. Without the "enable" privilege, only exec privilege level 1 commands are available to the remote user.

Additionally, remote users may use the remote copy (RCP) feature to transfer files from the router to the remote system (similar to HTTP and TFTP transfers). IOS may act as an rsh and rcp client itself, allowing for remote command execution and file transfers. The local hostname is always used as the source username, but the remote (target) username can be changed.

This configuration can be verified as follows.

```
Rack1R6#debug ip tcp rcmd
RCMD transactions debugging is on
Rack1R6#

Rack1R1#rsh 150.1.6.6 /user Rack1R6 show run interface Serial 0/0

Building configuration...

Current configuration : 110 bytes
!
interface Serial0/0
 ip address 54.1.1.6 255.255.255.0
 ip rip advertise 10
 encapsulation frame-relay
end

Rack1R6#
RCMD: [514 <- 150.1.1.1:988] recv \0
RCMD: [514 <- 150.1.1.1:988] recv Rack1R1\0Rack1R6\0show run interface
Serial 0/0\0
RCMD: [514 -> 150.1.1.1:988] send <OK>
```


Transfer a file from R6 to R1 using RCP

```
Rack1R6#show flash:

System flash directory:
File  Length   Name/status
  1   29631128  c2600-adventerprisek9-mz.124-10.bin
  2   1941     saved-config
  3   1036     r6i
  4   274807   crashinfo_20080727-230115
[29909172 bytes used, 3120968 available, 33030140 total]
32768K bytes of processor board System flash (Read/Write)

Rack1R1#copy rcp://150.1.6.6/saved-config null:
Source username [RCP]?
Accessing rcp://RCP@150.1.6.6/saved-config...!
1941 bytes copied in 0.056 secs (34661 bytes/sec)

Rack1R6#
RCMD: [514 <- 150.1.1.1:974] recv Rack1R1\0RCP\0rcp -f saved-config\0
RCMD: [514 -> 150.1.1.1:974] send <OK>
RCMD: [514 <- 150.1.1.1:974] recv <OK>
RCP: [514 -> 150.1.1.1:974] send C0644 1941 saved-config\n
RCMD: [514 <- 150.1.1.1:974] recv <OK>
RCP: [514 -> 150.1.1.1:974] send 1941 bytes
RCMD: [514 -> 150.1.1.1:974] send <OK>
RCMD: [514 <- 150.1.1.1:974] recv <BAD, Write failed>
RCMD: [514 <- 150.1.1.1:967] recv \0
RCMD: [514 <- 150.1.1.1:967] recv Rack1R1\0RCP\0rcp -f saved-config\0
RCMD: [514 -> 150.1.1.1:967] send <OK>
RCMD: [514 <- 150.1.1.1:967] recv <OK>
RCP: [514 -> 150.1.1.1:967] send C0644 1941 saved-config\n
RCMD: [514 <- 150.1.1.1:967] recv <OK>
RCP: [514 -> 150.1.1.1:967] send 1941 bytes
RCMD: [514 -> 150.1.1.1:967] send <OK>
RCMD: [514 <- 150.1.1.1:967] recv <OK>
```


12.29 NTP

- Configure R4 and R6 as authoritative NTP time sources with stratum 5.
- Both R4 and R6 should synchronize with each other.
- R5 should poll R4 and R6 for time updates, but prefer R4.
- Configure R5 to broadcast NTP updates on its connection to VLAN 58, and for SW2 to listen to NTP broadcasts on this link.
- Configure R6 to send multicast NTP packets to the address 239.1.1.1 on its connection to VLAN 67.
- SW1 and SW3 should be able to synchronize with R6 using these multicast packets.

Configuration

```
R4:
ntp master 5
ntp peer 150.1.6.6
ntp source Loopback0

R5:
ntp server 150.1.6.6
ntp server 150.1.4.4 prefer
ntp source Loopback0
!
interface FastEthernet 0/0
 ntp broadcast

R6:
ntp master 5
ntp peer 150.1.4.4
ntp source Loopback0

SW1:
ip multicast-routing distributed
!
interface Vlan67
 ip pim dense-mode
 ntp multicast client 239.1.1.1
!
interface Vlan79
 ip pim dense-mode

SW2:
interface Vlan58
 ntp broadcast client

SW3:
ip multicast-routing
!
interface Vlan79
 ip pim dense-mode
 ntp multicast client 239.1.1.1
```


Verification

Note

NTP time distribution is based on a loop-less tree topology. In the root of the tree resides the absolute time source (server), which is physically connected to an atomic clock, radio clock, or some other highly accurate time source. This server is described as being in *stratum 1*, meaning that it is one hop away from the source of the time.

Every next router that pulls time down from stratum N is automatically assigned to stratum N+1. That router in turn may become a time source (server) for clients in stratum N+2. A router in stratum N will not accept time updates from stratum N+1, enforcing a tree-level hierarchy of client-server relationships and preventing time-synchronization loops. If a router has multiple servers, it will only select one for time synchronization (the one which is closer to NTP root and has the least offset from the currently running clock) and use others as backup. The preference may be adjusted using the **`prefer`** keyword, but only among candidates having other parameters equal (e.g. same stratum).

Two routers in the same stratum may also form peer-to-peer, symmetric relationships. In this mode, both peers update each other's clock. However, just one router in pair may be configured for NTP peering (active peer), the other router will automatically convert into passive mode, and will provide time updates to the dynamic peer.

NTP is also capable of broadcasting or multicasting time updates on a shared interface, and clients may listen to NTP packets on the other side. This allows a single server to synchronize multiple hosts in unsolicited mode.

Remember that NTP may take considerably time to synchronize, especially if router clocks are far out of synch. To speed up this process adjust the clocks manually to values close to the server, in order to accelerate convergence (e.g. when router considers all servers as "insane").

Note that in the following show commands output reference “127.127.7.1” means “self”. Before beginning verifications, ensure you set all clocks to the same value manually. It may take some time still for both masters to synchronize and consider each other “sane”.

```
Rack1R4#clock set 00:00:01 Jan 1 2012
Rack1R5#clock set 00:00:01 Jan 1 2012
Rack1R6#clock set 00:00:01 Jan 1 2012
Rack1SW1#clock set 00:00:01 Jan 1 2012
Rack1SW2#clock set 00:00:01 Jan 1 2012
Rack1SW3#clock set 00:00:01 Jan 1 2012

Rack1R1#show ntp status
Clock is synchronized, stratum 5, reference is 127.127.7.1
nominal freq is 249.5901 Hz, actual freq is 249.5901 Hz, precision is
2**18
reference time is D2AA2FD6.B73C8C6D (01:05:26.715 UTC Sun Jan 1 2012)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 21.13 msec, peer dispersion is 21.13 msec

Rack1R4#show ntp associations

      address         ref clock     st  when  poll reach  delay  offset    disp
+~150.1.6.6        127.127.7.1       5    49    64  337     2.7  -28.16     3.7
*~127.127.7.1      127.127.7.1       4     9    64  377     0.0    0.00     0.0
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

Rack1R6#show ntp status
Clock is synchronized, stratum 5, reference is 127.127.7.1
nominal freq is 249.5901 Hz, actual freq is 249.5901 Hz, precision is 2**16
reference time is D2AA3016.B76908FA (01:06:30.716 UTC Sun Jan 1 2012)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 20.49 msec, peer dispersion is 20.49 msec

Rack1R6#show ntp associations

      address         ref clock     st  when  poll reach  delay  offset    disp
+~150.1.4.4        127.127.7.1       5    50    64  376     4.5   22.03     4.1
*~127.127.7.1      127.127.7.1       4    59    64  377     0.0    0.00     0.0
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

Rack1R5#show ntp status
Clock is synchronized, stratum 6, reference is 150.1.4.4
nominal freq is 249.5901 Hz, actual freq is 249.5903 Hz, precision is 2**16
reference time is D2AA3023.F4764E72 (01:06:43.954 UTC Sun Jan 1 2012)
clock offset is -6.2522 msec, root delay is 46.68 msec
root dispersion is 27.80 msec, peer dispersion is 1.65 msec
```



```
Rack1R5#show ntp associations

      address         ref clock     st  when  poll reach  delay  offset    disp
*~150.1.4.4        127.127.7.1       5    37    64  377    46.7   -6.25     1.6
+~150.1.6.6        127.127.7.1       5    26    64  377    47.0  -30.80     1.7
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

Rack1SW2#show ntp status
Clock is synchronized, stratum 7, reference is 155.1.58.5
nominal freq is 119.2092 Hz, actual freq is 119.2076 Hz, precision is 2**17
reference time is D2AA3058.E84973C0 (01:07:36.907 UTC Sun Jan 1 2012)
clock offset is -1.1060 msec, root delay is 48.45 msec
root dispersion is 905.29 msec, peer dispersion is 876.36 msec

Rack1SW2#show ntp associations

      address         ref clock     st  when  poll reach  delay  offset    disp
* 155.1.58.5       150.1.4.4         6    54    64  176     1.8   -1.11   876.4
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

Rack1SW3#show ip mroute 239.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 239.1.1.1), 00:35:37/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Vlan79, Forward/Dense, 00:35:37/00:00:00

(155.1.67.6, 239.1.1.1), 00:35:17/00:02:18, flags: PLTX
  Incoming interface: Vlan79, RPF nbr 155.1.79.7
  Outgoing interface list: Null

Rack1SW3#show ntp status
Clock is synchronized, stratum 6, reference is 155.1.67.6
nominal freq is 250.0000 Hz, actual freq is 249.9983 Hz, precision is 2**18
reference time is D2AA28D1.974FD2EF (00:35:29.591 UTC Sun Jan 1 2012)
clock offset is 3.3951 msec, root delay is 2.58 msec
root dispersion is 6.93 msec, peer dispersion is 3.51 msec
```


12.30 NTP Authentication

- Configure R4 and R6 to authenticate each others peering session using the key “CISCO46”.
- R5 should authenticate NTP packets received from R4 and R6 using key values “CISCO4” and “CISCO6” respectively.
- SW2 should only accept NTP updates on VLAN58 if they are authenticated using key value of “CISCO58”.

Configuration

```
R4:
ntp authenticate
ntp authentication-key 46 md5 CISCO46
ntp trusted-key 46
ntp peer 150.1.6.6 key 46
ntp authentication-key 4 md5 CISCO4

R5:
ntp authenticate
ntp authentication-key 4 md5 CISCO4
ntp authentication-key 6 md5 CISCO6
ntp trusted-key 4
ntp trusted-key 6
ntp server 150.1.4.4 key 4
ntp server 150.1.6.6 key 6
ntp authentication-key 58 md5 CISCO58
!
interface FastEthernet 0/0
 ntp broadcast key 58

R6:
ntp authenticate
ntp authentication-key 46 md5 CISCO46
ntp trusted-key 46
ntp peer 150.1.4.4 key 46
ntp authentication-key 6 md5 CISCO6

SW2:
ntp authenticate
ntp authentication-key 58 md5 CISCO58
ntp trusted-key 58
```


Verification

Note

NTP authentication is based on two concepts. First, NTP packets that can update (change) the local clock must be authenticated. Secondly, authenticated NTP packets are signed using an HMAC MD5 signature, and carry the signing key index (number) inside.

Therefore based on the above, only the router which updates its clock (e.g. client or peer) must be configured to require authentication. However, authentication keys must be configured on **both** sides (e.g. client and server) using the **same** key number (index). For example, if R1 is the NTP client and R2 is the NTP server, R1 may send authenticated (signed) NTP request to R2 with the key index 12. R2 will not verify the signature (it's not going to update its clock), but instead it simply looks up the key with index 12 in its local key chain. Using this key the NTP response is signed and sent back to R1.

In order to make router authenticate incoming updates, you must explicitly enable authentication, configure a key with key index, and ensure the authenticating router trusts this key. There is no need to trust keys which are **not** used to authenticate updates (e.g. reply keys).

```
Rack1R4#show ntp associations detail  | inc auth
150.1.6.6 configured, authenticated, selected, sane, valid, stratum 5

Rack1R6#show ntp associations detail  | inc auth
150.1.4.4 configured, authenticated, selected, sane, valid, stratum 5

Rack1R5#show ntp associations detail | inc auth
150.1.4.4 configured, authenticated, selected, sane, valid, stratum 5
150.1.6.6 configured, authenticated, our_master, sane, valid, stratum 5

Rack1SW2#show ntp associations detail | inc auth
155.1.58.5 dynamic, authenticated, our_master, sane, valid, stratum 6
```


12.31 NTP Access Control

- Configure R4 and R6 so that no other devices can update their clocks.
- R5 should only allow R4 and R6 to update its clock.
- R4 and R6 should not serve time out to other devices besides R5.
- SW2 should only accept time from R5 and not other sources, but R5 should be allowed to serve time out to other devices.

Configuration

```
R4:
access-list 6 permit 150.1.6.6
access-list 6 permit 127.127.7.1
access-list 5 permit 150.1.5.5
!
ntp access-group peer 6
ntp access-group serve-only 5

R5:
access-list 46 permit 150.1.4.4
access-list 46 permit 150.1.6.6
!
ntp access-group peer 46

R6:
access-list 4 permit 150.1.4.4
access-list 4 permit 127.127.7.1
access-list 5 permit 150.1.5.5
!
ntp access-group peer 4
ntp access-group serve-only 5

SW2:
access-list 58 permit 155.1.58.5
!
ntp access-group peer 58
```


Verification

Note

NTP access-control divides NTP messages in two categories, control messages and update/request messages. Control messages are those needed to extract specific management information, such as the peer status or set a management parameter. NTP control messages are not needed for proper time synchronization. NTP update/request messages are those messages needed for time synchronization.

NTP access control defines four levels, peer, serve, serve-only, and query-only. Peer permits NTP updates/requests to the host as well as control queries. This is the only access type which allows the host to be synchronized by others. Serve permits NTP requests, but rejects NTP updates (does not change local clock). Control queries are also permitted. Serve-only permits NTP requests only. It rejects attempts to synchronize the local system and does not accept control queries. Query-only accepts NTP control queries. No response to NTP requests are sent, and no local system time synchronization with a remote system is allowed.

When these levels are associated with access-lists on a router an incoming message source IP address is matched in the order listed above (peer, serve, serve-only, and query-only). The first match determines the type of access. If some access-groups are configured but not all, all other types of access from any other sources are implicitly denied.

Note that in this configuration the masters must allow the NTP process to synchronize with themselves, the peer “127.127.7.1”. This is only needed since they have a “peer” access-group for other sources, which controls who can update the clock.

To verify the access-groups change R5's VLAN 5 IP address and see how SW2 is affected.

```
Rack1SW2#show ntp associations detail
155.1.58.5 dynamic, authenticated, our_master, sane, valid, stratum 6
ref ID 150.1.6.6, time D2AA384E.ECE8E83D (01:41:34.925 UTC Sun Jan 1 2012)
our mode bdcast client, peer mode bdcast, our poll intvl 64, peer poll intvl 64
root delay 57.43 msec, root disp 25.04, reach 376, sync dist 59.677
delay 1.77 msec, offset 4.2912 msec, dispersion 5.04
precision 2**16, version 3
org time D2AA3858.DE49674B (01:41:44.868 UTC Sun Jan 1 2012)
rcv time D2AA3858.DD302D03 (01:41:44.864 UTC Sun Jan 1 2012)
xmt time 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
filtdelay =     1.77    1.77    1.77    1.77    1.77    1.77    1.77    1.77
filtoffset =    4.29    0.98    0.18   -1.07   -1.43   -1.69   -1.91   -2.21
filterror =     0.99    1.97    2.94    3.92    4.90    5.87    6.85    7.83

Rack1R5#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R5(config)#interface fastEthernet 0/0
Rack1R5(config-if)#ip address 155.1.58.55 255.255.255.0
Rack1R5(config-if)#
```

Note that although SW2 receives packets from R5's new IP address it does not accept it as a new dynamic server. The old information is kept until it expires.

```
Rack1SW2#debug ntp packet
Rack1SW2#
NTP: rcv packet from 155.1.58.55 to 255.255.255.255 on Vlan58:
 leap 0, mode 5, version 3, stratum 6, ppoll 64
 rtdel 0E97 (56.992), rtdsp 0663 (24.948), refid 96010606 (150.1.6.6)
 ref D2AA398E.EDCC3A82 (01:46:54.928 UTC Sun Jan 1 2012)
 org 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
 rec 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)
 xmt D2AA3998.DF41F31B (01:47:04.872 UTC Sun Jan 1 2012)
 inp D2AA3998.DCA90836 (01:47:04.861 UTC Sun Jan 1 2012)

Rack1SW2#show ntp associations

      address         ref clock     st  when  poll reach  delay  offset    disp
* 155.1.58.5       150.1.6.6         6   307    64  340     1.8    5.39  16000.
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured
```


12.32 Auto-Install over LAN Interfaces using DHCP

- Configure the network to support auto-install as follows:
  - Upon booting R4 should request an IP address for its FastEthernet0/1 interface via DHCP.
  - R1 should forward this request onto R5.
  - R5 should reply with the IP address 155.X.146.4/24, the default gateway 155.X.146.1, the DNS server 155.X.146.6, and the TFTP server “RackXSW2”.
  - R4 should then ask R6 what its own hostname is, and download the config file “RackXR4-confg” from SW2 via TFTP.

Configuration

```
R6:
ip dns server
ip host Rack1SW2 155.1.58.8
ip host Rack1R4 155.1.146.4

R1:
interface FastEthernet 0/0
 ip helper-address 155.1.0.5

R5:
ip dhcp pool HOST_R4
   host 155.1.146.4 255.255.255.0
   client-identifier
0063.6973.636f.2d30.3030.372e.6562.6465.2e35.3632.322d.4574.302f.31
   default-router 155.1.146.1
   dns-server 155.1.146.6
   option 66 ascii "Rack1SW2"

SW2:
tftp-server flash:Rack1R4-confg
```


Verification

Note

AutoInstall tries to obtain an IP address and configuration by broadcasting DHCP/BOOTP/RARP requests over all LAN interfaces, using HDLC encapsulation on the first serial interface and the SLARP protocol to obtain an IP address, and by using Frame-Relay encapsulation on first serial interface and the BOOTP protocol to obtain an IP address.

On regular Ethernet LAN segments the AutoInstall process relies primarily on DHCP for obtaining an IP address and TFTP for downloading the configuration. The procedure is performed in the following stages.

First the router tries to obtain an IP address and subnet mask via DHCP or BOOTP by broadcasting DHCP/BOOTP requests out of all LAN interfaces. In addition to IP address information, the DHCP server can provide the default-router, DNS server list, TFTP server, and a boot-file name. In parallel to DHCP, IOS sends BOOTP queries, and may the learn TFTP server's name ("sname") or IP address ("siaddr") and boot-file name ("file") values from BOOTP.

Next the router tries to download its configuration using unicast transfer from the TFTP server learned via DHCP. If the DHCP/BOOTP server has specified the boot-file, the router tries to download this file as its configuration. Prior to version 12.1(5)T IOS only tried broadcast requests to reach the TFTP server and ignored the boot-file name learned via DHCP/BOOTP using generic names as described later.

Next, for the destination IP address of the TFTP server, the router either uses Option 150 or tries to resolve Option 66 using the DNS server list supplied by DHCP. If a BOOTP response is available with the "sname" and "siaddr" fields they could also be used. The order of preference is: "sname", "option 66", "option 150", and "siaddr". In case if the TFTP server's IP address could not be obtained, the router broadcasts download requests to the TFTP port. If the TFTP server does not respond to three unicast requests, the router falls back to broadcasts.

If the DHCP/BOOTP server did not provide a boot-file name, the router tries to download a network-wide configuration file named “network-confg” and then “cisconet.cfg” from the TFTP server (using either unicasts if the IP address of the TFTP server is known, or broadcasts). The purpose of network-wide configuration file is to allow the router to determine its hostname.

If the network-wide configuration has been successfully downloaded, the router looks into the **`ip host`** commands trying to find a match to its newly obtained IP address and map it to a hostname. If the hostname is successfully found, the router requests the host-specific configuration file “<hostname>-confg” via TFTP, and if this fails it requests “<hostname>.cfg”. If the file has been downloaded, it is merged with the router’s configuration.

If the hostname has not been determined the router tries to resolve its name via DNS (using either DNS servers or broadcasts) and downloads the respective configuration file. If the name lookup or host-specific configuration file download has failed, the router tries to download the generic configuration file named either “router-confg” or “router.cfg”

To verify this copy R4’s configuration to SW2 via TFTP, then erase and reload R4.

```
Rack1R4#copy running-config flash:Rack1R4-confg
Destination filename [Rack1R4-confg]?
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0xC982)
1459 bytes copied in 8.033 secs (182 bytes/sec)

Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#tftp-server flash:Rack1R4-confg

Rack1SW2#copy tftp: flash:
Address or name of remote host []? 150.1.4.4
Source filename []? Rack1R4-confg
Destination filename [Rack1R4-confg]?
Accessing tftp://150.1.4.4/Rack1R4-confg...
Loading Rack1R6-confg from 150.1.4.4 (via Vlan58): !
[OK - 1459 bytes]
```


Now find out the Client-ID of R4's FastEthernet 0/1 interface so we know what client-identifier to put in R5's DHCP host pool. To accomplish this temporarily set up the respective interface for DHCP IP address assignment and use **show dhcp lease** command.

```
Rack1R4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#interface FastEthernet0/1
Rack1R4(config-if)#ip address dhcp
Rack1R4(config-if)#end
Rack1R4#show dhcp lease
Temp IP addr: 0.0.0.0  for peer on Interface: FastEthernet0/1
Temp  sub net mask: 0.0.0.0
   DHCP Lease server: 0.0.0.0, state: 1 Selecting
   DHCP transaction id: 14F4
   Lease: 0 secs,  Renewal: 0 secs,  Rebind: 0 secs
   Next timer fires after: 00:00:01
   Retry count: 0   Client-ID: cisco-0007.ebde.5622-Et0/1
   Client-ID hex dump: 636973636F2D303030372E656264652E
                       353632322D4574302F31
   Hostname: Rack1R4

Rack1R4#show interfaces FastEthernet 0/1
FastEthernet0/1 is up, line protocol is up
  Hardware is AmdP2, address is 0007.ebde.5622 (bia 0007.ebde.5622)
```

Cisco uses the MAC address of the interface along with interface short name to create a client-identifier. In the Cisco IOS DHCP server the client-ID must be specified as hex value, starting with '00' (e.g. 0063.6973.636F.2D30.3030.372E.6562.6465.2E35.3632.322D.4574.302F.31).

When R4 is reset the DHCP request is received by R5, and the TFTP request is received by SW2.

```
Rack1R4#show start
startup-config is not present
Rack1R4#reload
Proceed with reload? [confirm]

SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload
Command.

Rack1R5#debug  ip dhcp server events
Rack1R5#debug  ip dhcp server packet

Rack1SW2#debug tftp events
TFTP Event debugging is on

Rack1SW2#debug tftp packets
TFTP Packet debugging is on
```



```
Rack1R5#
DHCPD: DHCPDISCOVER received from client
0063.6973.636f.2d30.3030.372e.6562.6465.2e35.3632.322d.4574.302f.31
through relay 155.1.146.1.
DHCPD: Seeing if there is an internally specified pool class:
  DHCPD: htype 1 chaddr 0007.ebde.5622
  DHCPD: circuit id 01f50000
DHCPD: Sending DHCPOFFER to client
0063.6973.636f.2d30.3030.372e.6562.6465.2e35.3632.322d.4574.302f.31
(155.1.146.4).
DHCPD: unicasting BOOTREPLY for client 0007.ebde.5622 to relay
155.1.146.1.
DHCPD: DHCPREQUEST received from client
0063.6973.636f.2d30.3030.372e.6562.6465.2e35.3632.322d.4574.302f.31.
DHCPD: Sending notification of ASSIGNMENT:
 DHCPD: address 155.1.146.4 mask 255.255.255.0
  DHCPD: lease time remaining (secs) = 4294967295
DHCPD: No default domain to append - abort update
DHCPD: Sending DHCPACK to client
0063.6973.636f.2d30.3030.372e.6562.6465.2e35.3632.322d.4574.302f.31
(155.1.146.4).
DHCPD: unicasting BOOTREPLY for client 0007.ebde.5622 to relay
155.1.146.1.

Rack1SW2#
TFTP: read request from host 155.1.146.4(56461) via Vlan58
TFTP: Opened flash:Rack1R4-confg, fd 0, size 1494
TFTP: Sending block 1 (retry 0), socket_id 0x2373918
TFTP: Received ACK for block 1, socket_id 0x2373918
TFTP: Sending block 2 (retry 0), socket_id 0x2373918
TFTP: Received ACK for block 2, socket_id 0x2373918
TFTP: Sending block 3 (retry 0), socket_id 0x2373918
TFTP: Received ACK for block 3, socket_id 0x2373918
TFTP: Finished flash:Rack1R4-confg, time 00:00:00
TFTP: Server request for port 54974, socket_id 0x3B89C38
TFTP: read request from host 155.1.146.4(54974) via Vlan58
TFTP: Opened flash:Rack1R4-confg, fd 0, size 1494
TFTP: Sending block 1 (retry 0), socket_id 0x3B89C38
TFTP: Received ACK for block 1, socket_id 0x3B89C38
TFTP: Sending block 2 (retry 0), socket_id 0x3B89C38
TFTP: Received ACK for block 2, socket_id 0x3B89C38
TFTP: Sending block 3 (retry 0), socket_id 0x3B89C38
TFTP: Received ACK for block 3, socket_id 0x3B89C38
TFTP: Finished flash:Rack1R4-confg, time 00:00:00
```



```
R4:
32K bytes of NVRAM.
32768K bytes of processor board System flash (Read/Write)

Translating "Rack1SW2"...domain server (155.1.146.6) [OK]

%Error opening tftp://155.1.58.8/network-confg (No such file or directory)
%Error opening tftp://155.1.58.8/cisconet.cfg (No such file or directory)
Domain server mapped address 155.1.146.4 to Rack1R4
Loading rack1r4-confg from 155.1.58.8 (via FastEthernet0/1): !
[OK - 1358 bytes]
Configuration mapped ip address 155.1.58.8 to Rack1SW2
 Slot is empty or does not support clock participate
 WIC slot is empty or does not support clock participate
%Error opening tftp://255.255.255.255/network-confg (Timed out)
%Error opening tftp://255.255.255.255/cisconet.cfg (Timed out)

         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no

Press RETURN to get started!

<snip>
*Mar  1 00:01:15.215: AUTOINSTALL: FastEthernet0/1 is assigned 155.1.146.4
*Mar  1 00:01:15.215: AUTOINSTALL: Obtain tftp server name Rack1SW2resolved to
155.1.58.8
*Mar  1 00:01:18.288: AUTOINSTALL: Obtain default router (opt 3) 155.1.146.1

<snip>
```

Rack1R4#show ip int brief

```
Interface                  IP-Address      OK? Method Status          Protocol
FastEthernet0/0            204.12.1.4      YES TFTP   up                    up
Serial0/0                  unassigned      YES TFTP   up                    up
Serial0/0.1                155.1.0.4       YES TFTP   up                    up
FastEthernet0/1            155.1.146.4     YES TFTP   up                    up
Serial0/1                  155.1.45.4      YES TFTP   up                    up
Loopback0                  150.1.4.4       YES TFTP   up                    up
```


12.33 Auto-Install over Frame-Relay

- Disable R1's link to VLAN 146.
- Configure R5 as a staging router for AutoInstall over Frame-Relay for R4 as follows:
    - R5 should assign R4 the IP address 155.X.0.4.
    - When R5 receives a TFTP request from R4 for the file "network-confg" it should be forwarded to SW2.
    - R4 should download this config from SW2 and learn its hostname RackXR4.
    - R4 should then download the config file "RackXR4-confg" from SW2.

Configuration

```
R1:
inteface FastEthernet 0/0
 shutdown

R5:
interface Serial 0/0
 ip helper-address 155.1.58.8

SW2:
tftp-server flash:rack1r4-confg
tftp-server flash:network-confg
```

Verification

Note

Auto-Install over Frame-Relay is tried after auto-install attempts on all LAN interfaces and after HDLC/SLARP has been tried. As with all auto-install attempts over Serial interfaces, only the first serial interface of the router is tried.

On the first try, the encapsulation used is HDLC and SLARP requests are sent to the other side. If this fails, Frame-Relay encapsulation is configured on the first serial interface (e.g. Serial0 or Serial 1/0), and BOOTP broadcast packets are sent on all active DLCIs. The source IP address used in BOOTP packets is "0.0.0.0". The staging router receiving BOOTP packets on the other side of the Frame-Relay cloud will generate BOOTP reply based on its own **`frame-relay map`** commands. In this case R5 sees a BOOTP requests arrive on DLCI 504, and this DLCI is mapped to the IP address 155.X.0.4. This causes R5 to send a BOOTP reply with the IP address 155.X.0.4.

After the IP address has been obtained on the Frame-Relay connection, the router broadcasts TFTP requests out of all available DLCIs using its newly obtained IP address as a source. The staging router should either forward those requests to a TFTP server using the **ip helper-address** or reply to the TFTP requests itself.

The new router will request the network-wide configuration file, followed by host-specific configuration file as described in the previous scenario.

Follow the procedure below to create network-wide and host-specific configuration files, and upload them to SW2. The files differ in that the network-specific file contains hostname to IP address mapping for R4.

```
Rack1R4#copy running-config flash:Rack1R4-confg
Destination filename [Rack1R4-confg]?
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0xC982)
1459 bytes copied in 8.033 secs (182 bytes/sec)

Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#tftp-server flash:Rack1R4-confg

Rack1SW2#copy tftp: flash:
Address or name of remote host []? 150.1.4.4
Source filename []? Rack1R4-confg
Destination filename [Rack1R4-confg]?
Accessing tftp://150.1.4.4/Rack1R4-confg...

Rack1R4(config)#ip host Rack1R4 155.1.0.4
Rack1R4(config)#end
Rack1R4#copy running-config flash:Rack1R4-confg
Destination filename [network-confg]?
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0xC982)
1459 bytes copied in 8.033 secs (182 bytes/sec)

Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#tftp-server flash:network-confg

Rack1SW2#copy tftp: flash:
Address or name of remote host []? 150.1.4.4
Source filename []? network-confg
Destination filename [network-confg]?
Accessing tftp://150.1.4.4/network-confg...
```


To ensure SW2 exports both configuration files via TFTP and erase R4's configuration and reload the router. Observe the debugging output on R5 and SW2 to see the auto-configuration process.

```
Rack1R5#debug  ip dhcp server events
Rack1R5#debug  ip dhcp server packet

Rack1SW2#debug tftp events
TFTP Event debugging is on

Rack1SW2#debug tftp packets
TFTP Packet debugging is on

Rack1R5#
DHCPD: BOOTREQUEST received from BOOTP client 0007.ebde.5621 on
interface Serial0/0.
DHCPD: there is no address pool for 155.1.0.5.

Rack1SW2#
TFTP: Server request for port 49515, socket_id 0x3B89420
TFTP: read request from host 155.1.0.4(49515) via Vlan58
TFTP: Opened flash:network-confg, fd 0, size 1361
TFTP: Sending block 1 (retry 0), socket_id 0x3B89420
TFTP: Received ACK for block 1, socket_id 0x3B89420
TFTP: Sending block 2 (retry 0), socket_id 0x3B89420
TFTP: Received ACK for block 2, socket_id 0x3B89420
TFTP: Sending block 3 (retry 0), socket_id 0x3B89420
TFTP: Received ACK for block 3, socket_id 0x3B89420
TFTP: Finished flash:network-confg, time 00:00:00
<snip>
TFTP: Opened flash:Rack1R4-confg, fd 0, size 1494
TFTP: Sending block 1 (retry 0), socket_id 0x3989B80
TFTP: Received ACK for block 1, socket_id 0x3989B80
TFTP: Sending block 2 (retry 0), socket_id 0x3989B80
TFTP: Received ACK for block 2, socket_id 0x3989B80
TFTP: Sending block 3 (retry 0), socket_id 0x3989B80
TFTP: Received ACK for block 3, socket_id 0x3989B80
TFTP: Finished flash:Rack1R4-confg, time 00:00:00
<snip>
```


12.34 Auto-Install over LAN Interfaces using RARP

- Configure R1 to supply an IP address to R4 via RARP on VLAN 146 when it boots.
- SW2 should supply the network and host-specific configuration files to R4.

Configuration

```
R1:
!
! ARP entry for R4's VLAN 146 interface
!
arp 155.1.146.4 0007.ebde.5622 arpa
!
inteface FastEthernet 0/0
 no shutdown
 ip rarp-server 155.1.146.1
 ip helper-address 155.1.58.8

SW2:
tftp-server flash:rack1r4-confg
tftp-server flash:network-confg
```


Verification

Note

RARP is a simple alternative to using DHCP/BOOTP for IP address configuration on the LAN interface. A router sends out RARP requests for an IP address after it fails to obtain an address via DHCP/BOOTP. Another IOS router could be configured as RARP server, responding to a MAC-to-IP mapping request from a configured static ARP entry. After obtaining an IP address the router may proceed with TFTP broadcasts for network and host specific configuration files. Another router on the same subnet could relay the requests to the TFTP server.

Follow the procedure below to create a network-wide and route-specific configuration files and upload them to SW2. The files differ in that network-specific file contains hostname to IP address mapping for R4, as in the previous example.

```
Rack1R4#copy running-config flash:Rack1R4-confg
Destination filename [Rack1R4-confg]?
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0xC982)
1459 bytes copied in 8.033 secs (182 bytes/sec)

Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#tftp-server flash:Rack1R4-confg

Rack1SW2#copy tftp: flash:
Address or name of remote host []? 150.1.4.4
Source filename []? Rack1R4-confg
Destination filename [Rack1R4-confg]?
Accessing tftp://150.1.4.4/Rack1R4-confg...

Rack1R4(config)#no ip host Rack1R4
Rack1R4(config)#ip host Rack1R4 155.1.146.4
Rack1R4(config)#end
Rack1R4#copy running-config flash:Rack1R4-confg
Destination filename [network-confg]?
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0xC982)
1459 bytes copied in 8.033 secs (182 bytes/sec)

Rack1R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Rack1R4(config)#tftp-server flash:network-confg

Rack1SW2#copy tftp: flash:
Address or name of remote host []? 150.1.4.4
Source filename []? network-confg
Destination filename [network-confg]?
Accessing tftp://150.1.4.4/network-confg...
```


To ensure SW2 exports both configuration files via TFTP erase R4's configuration and reload the router. Observe the debugging output on R1 and SW2 to see the auto-configuration process.

```
Rack1SW2#debug tftp events
TFTP Event debugging is on

Rack1SW2#debug tftp packets
TFTP Packet debugging is on

Rack1R1#debug arp
ARP packet debugging is on

Rack1R1#
RARP: Rcvd RARP req for 0007.ebde.5622
RARP: Sending reply out FastEthernet0/0
IP RARP: sent rev-rep src 155.1.146.1 000d.bc04.e340,
                 dst 155.1.146.4 0007.ebde.5622 FastEthernet0/0

Rack1SW2#
TFTP: read request from host 155.1.146.4(54757) via Vlan58
TFTP: Opened flash:network-confg, fd 0, size 1440
TFTP: Sending block 1 (retry 0), socket_id 0x3B89500
TFTP: Received ACK for block 1, socket_id 0x3B89500
TFTP: Sending block 2 (retry 0), socket_id 0x3B89500
TFTP: Received ACK for block 1, socket_id 0x3B89500
TFTP: Sending block 2 (retry 1), socket_id 0x3B89500
TFTP: Received ACK for block 2, socket_id 0x3B89500
TFTP: Sending block 3 (retry 0), socket_id 0x3B89500
TFTP: Received ACK for block 2, socket_id 0x3B89500
TFTP: Sending block 3 (retry 1), socket_id 0x3B89500
TFTP: Received ACK for block 3, socket_id 0x3B89500
TFTP: Finished flash:network-confg, time 00:00:08
<snip>
TFTP: Opened flash:Rack1R4-confg, fd 0, size 1494
TFTP: Sending block 1 (retry 0), socket_id 0x3989B80
TFTP: Received ACK for block 1, socket_id 0x3989B80
TFTP: Sending block 2 (retry 0), socket_id 0x3989B80
TFTP: Received ACK for block 2, socket_id 0x3989B80
TFTP: Sending block 3 (retry 0), socket_id 0x3989B80
TFTP: Received ACK for block 3, socket_id 0x3989B80
TFTP: Finished flash:Rack1R4-confg, time 00:00:00
<snip>
```


12.35 IOS Menus

- When a user named OPERATOR telnets into R1 and authenticates with the password of CISCO it should see the following menu:

```
Operator Menu


    1          Display IP Routing Table

    2          Display Running Config

    3          Escape to Shell

    4          Disconnect
```

- Option 1 should show the routing table.
- Option 2 should show the running configuration.
- Option 3 should exit to the exec prompt.
- Option 4 should exit the telnet session.
- Clear the screen when the menu starts.

Configuration

```
R1:
menu OPERATOR title #
Operator Menu
#
menu OPERATOR text 1 Display IP Routing Table
menu OPERATOR command 1 show ip route
menu OPERATOR text 2 Display Running Config
menu OPERATOR command 2 show run
menu OPERATOR text 3 Escape to Shell
menu OPERATOR command 3 menu-exit
menu OPERATOR text 4 Disconnect
menu OPERATOR command 4 exit
menu OPERATOR clear-screen
!
username OPERATOR autocommand menu OPERATOR
username OPERATOR password CISCO
username OPERATOR privilege 15
!
line vty 0 4
 login local
```


Verification

Note

Menus allow simple automation of many network-monitoring tasks for unskilled/untrusted personnel. Each menu selection is associated with a command which is executed when a user chooses the selection. The command is executed with privileges of the local user who executed the menu command, or the privileges of the terminal line if the menu command is associated with a line. Therefore if the menu uses privileged commands (e.g. level 15), ensure they are available in the privilege level associated with the user/line.

The special menu command **menu-exit** allows a user to leave the menu but to not disconnect from the line. That means once this command is executed the user is put back into the exec shell.

```
Rack1R1#telnet 150.1.1.1
Trying 150.1.1.1 ... Open


User Access Verification

Username: OPERATOR
Password: CISCO

Operator Menu


    1           Display IP Routing Table

    2           Display Running Config

    3           Escape to Shell

    4           Disconnect


Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R    204.12.1.0/24 [120/1] via 155.1.146.4, 00:00:06, FastEthernet0/0
     155.1.0.0/24 is subnetted, 15 subnets
C       155.1.146.0 is directly connected, FastEthernet0/0
R       155.1.23.0 [120/1] via 155.1.13.3, 00:00:06, Serial0/1
R       155.1.10.0 [120/3] via 155.1.0.5, 00:00:01, Serial0/0
R       155.1.8.0 [120/2] via 155.1.0.5, 00:00:01, Serial0/0
```



```
R       155.1.9.0 [120/3] via 155.1.146.6, 00:00:02, FastEthernet0/0
                  [120/3] via 155.1.13.3, 00:00:06, Serial0/1
C       155.1.13.0 is directly connected, Serial0/1
C       155.1.0.0 is directly connected, Serial0/0
R       155.1.7.0 [120/2] via 155.1.146.6, 00:00:02, FastEthernet0/0
                  [120/2] via 155.1.13.3, 00:00:06, Serial0/1

Operator Menu


    1           Display IP Routing Table

    2           Display Running Config

    3           Escape to Shell

    4           Disconnect


Building configuration...

Current configuration : 1989 bytes
!
! Last configuration change at 23:15:46 UTC Tue Jul 29 2008
! NVRAM config last updated at 22:03:31 UTC Tue Jul 29 2008
!
version 12.4
service timestamps debug datetime msec
<snip>

Operator Menu


    1           Display IP Routing Table

    2           Display Running Config

    3           Escape to Shell

    4           Disconnect


Rack1R1#
```


12.36 IOS Banners

- Change the default IOS banners on R3 as follows:
  - All connecting users, with the exception to console users, should see the notification message “Welcome to IOS Router”.
  - Before the login prompt is displayed users should see the message “Please, authenticate yourself”.
  - Before a user can see the shell prompt, display “Hi, you are on the line <Line> have a nice time at <Hostname>” where <Line> is the line number the user is connected to and <Hostname> is the current router’s hostname.
  - Set the banner for reverse telnet connections to “This is a reverse telnet connection”.

Configuration

```
R3:
banner motd #
Welcome to IOS Router
#
banner login #
Please authenticate yourself
#
banner exec #
Hi, you are on the line $(line), have a nice time at $(hostname)
#
banner incoming #
This is a reverse telnet connection
#
line console 0
 no motd-banner
 no exec-banner
```


Verification

Note

For remote access and local console connections the following types of banners can be displayed:

1) MOTD banner – displayed first to everyone connecting to the router
2) Login banner – displayed before the authentication prompt is presented to the user (if login is disabled, the banner is not displayed)
3) Exec-banner – displayed before the exec prompt is displayed to the user
4) Incoming-banner – displayed to users connecting via reverse telnet

Special tokens can be used the banner messages such as $(line) or $(hostname) to create dynamic notifications. Some lines can be selectively exempted from viewing banner messages, such as the console is in this example.

```
Rack1R3#telnet 150.1.3.3
Trying 150.1.3.3 ... Open

Welcome to IOS Router

Please authenticate yourself


User Access Verification

Password: cisco
Hi, you are on the line 130, have a nice time at Rack1R3

Rack1R3>
```

Login to R3 via the console and type **exit**. Login again to check the banners. Note that only the login banner is displayed.

```
Rack1R3#exit

<snip>

Welcome to IOS
```


Simulate a reverse telnet connection to R3's AUX port to verify the incoming-banner message. To do so, configure the AUX line to accept incoming telnet connections. In the below example the AUX port is line 129.

```
R3:
line aux 0
 no modem inOut
 transport input telnet

Rack1R3# show line
   Tty Typ     Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
*    0 CTY               -    -      -    -    -      6       0     0/0       -
   129 AUX   9600/9600   -    -      -    -    -      1       0     0/0       -
   130 VTY               -    -      1    -    -      5       0     0/0       -
   131 VTY               -    -      -    -    -      0       0     0/0       -
   132 VTY               -    -      -    -    -      0       0     0/0       -
   133 VTY               -    -      -    -    -      0       0     0/0       -
   134 VTY               -    -      -    -    -      0       0     0/0       -

Line(s) not in async mode -or- with no hardware support:
1-128

Rack1R3#telnet 150.1.3.3 6129
Trying 150.1.3.3, 6129 ... Open

Welcome to IOS Router

This is a reverse telnet connection
```

12.37 KRON Command Schedule

- Configure a KRON occurrence on R4 that saves the running config to a remote TFTP server daily at 08:00.
- Use the server 155.1.146.100 and the filename “r3-config”

Configuration

```
R3:
kron occurrence SAVE_DAILY at 8:00 recurring
policy-list SAVE_CONFIG
!
kron policy-list SAVE_CONFIG
cli show running-config | redirect tftp://155.1.146.100/r3-config
```

Verification

> **Note**
>
> The KRON Command Scheduler allows exec commands or TCL scripts to run at a specified time. The KRON configuration consists of a policy-list containing exec commands, and a scheduler to execute the commands in the policy-list at a specified time or recurring interval.

```
Rack1R3#debug kron all
All kron debug flags are on

Rack1R3#clock set 07:59:59 Jan 1 2010
Jan  1 07:59:59.000: Clock Set Seen
Jan  1 07:59:59.000: %SYS-6-CLOCKUPDATE: System clock has been updated
from 08:00:21 UTC Mon Jan 1 2001 to 07:59:59 UTC Fri Jan 1 2010,
configured from console by console.
Jan  1 07:59:59.000: Major 4, Minor 9
Jan  1 07:59:59.000: Start timer for SAVE_DAILY, expires in 60000 Msec

Rack1R3#show kron schedule
Kron Occurrence Schedule
SAVE_DAILY inactive, will run again in 0 days 00:00:05 at 8 :00 on

Rack1R3#show kron schedule
Kron Occurrence Schedule
SAVE_DAILY   active, will run again in 0 days 23:59:59 at 8 :00 on

Jan  1 08:00:59.002: Major 1, Minor 0
Jan  1 08:00:59.002: Timer Event SAVE_DAILY
Jan  1 08:00:59.002: Kron delay for next SAVE_DAILY 86400000
Jan  1 08:00:59.006: Call parse_cmd 'show running-config | redirect
tftp://155.1.146.100/r3-config'
Jan  1 08:01:24.083: Kron CLI return 0
<snip>
```


