Basic IOS Security Configuration

The following lessons and case studies are dedicated to basic Cisco IOS Software security configuration methods and are grouped into several scenarios, variations of which you are likely to encounter in the CCIE Security lab exam or in real life.

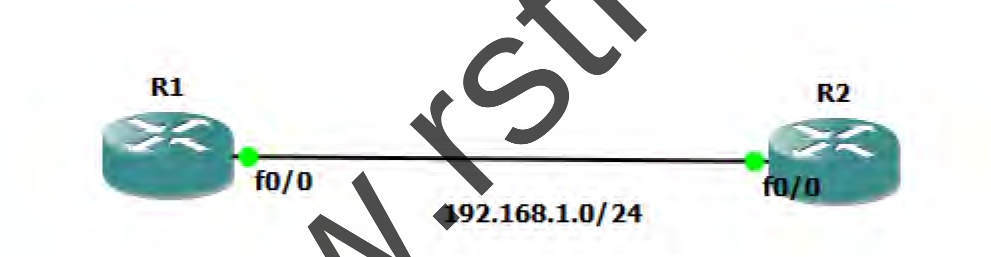
Lesson 15-1: Configuring Passwords, Privileges, and Logins

In this lesson, R2 is the router that needs to have basic Cisco IOS Software security features configured. Once R2 is configured, a remote host attempts to log in and perform some tasks.

This lesson covers the following configuration steps:

Step 1 Setting passwords
Step 2 Limiting connection time
Step 3 Configuring vtys and accessing the network remotely
Step 4 Creating user accounts
Step 5 Assigning privileges
Step 6 Local authentication, authorization, and accounting
Step 7 Remote administration with FTP
Step 8 Hiding Telnet addresses
Step 9 Verification

Topology:



Step 1: Setting Passwords

First, you have to protect access to a router by setting various passwords. Prevent unauthorized login by configuring passwords on the console and virtual terminal lines. The syntax for both of them is identical, as follows:

```
R3(config-line)#password string
```

After the line passwords are set, you need to take care of the privileged EXEC level. You should not use the **enable password** command because it is not secure and can give away a system password. Instead, opt for the following command:

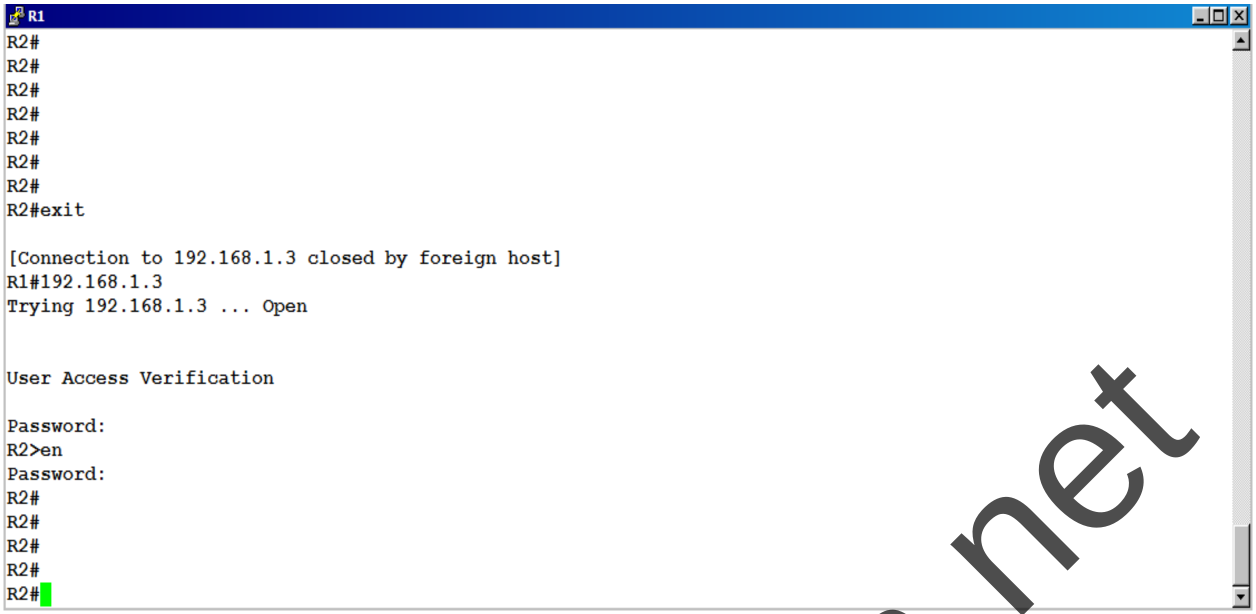
```
R3(config)#enable secret string
```

The **enable secret** command, as well as the username passwords described in "Creating User Accounts," later in this lesson, can be up to 25 characters long, including spaces, and are case sensitive. Example 15-1 demonstrates the application of passwords on R3. Note that both the console and the vty passwords appear scrambled. This is because **service password-encryption** is enabled on the router to hide the real string from a passerby.

Cofiguration:

```
version 12.4
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$36h1$rJTseJncrJCshy7ry3.zB1
!
line con 0
 exec-timeout 0 0
 privilege level 15
 password 7 00171B0F084B0A
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
```

Verification:



```
R2#
R2#
R2#
R2#
R2#
R2#
R2#
R2#exit

[Connection to 192.168.1.3 closed by foreign host]
R1#192.168.1.3
Trying 192.168.1.3 ... Open


User Access Verification

Password:
R2>en
Password:
R2#
R2#
R2#
R2#
R2#
```

Step 2: Limiting Connection Time

For security reasons, you do not want to leave the connection to any port, be it console or remote connection, logged in indefinitely. If the connections are configured to time out automatically, the administrator is logged out by a router after a specified period if he forgets to do it himself. The syntax is the same for any line and is as follows:

R3(config-line)#**exec-timeout** *minutes seconds*

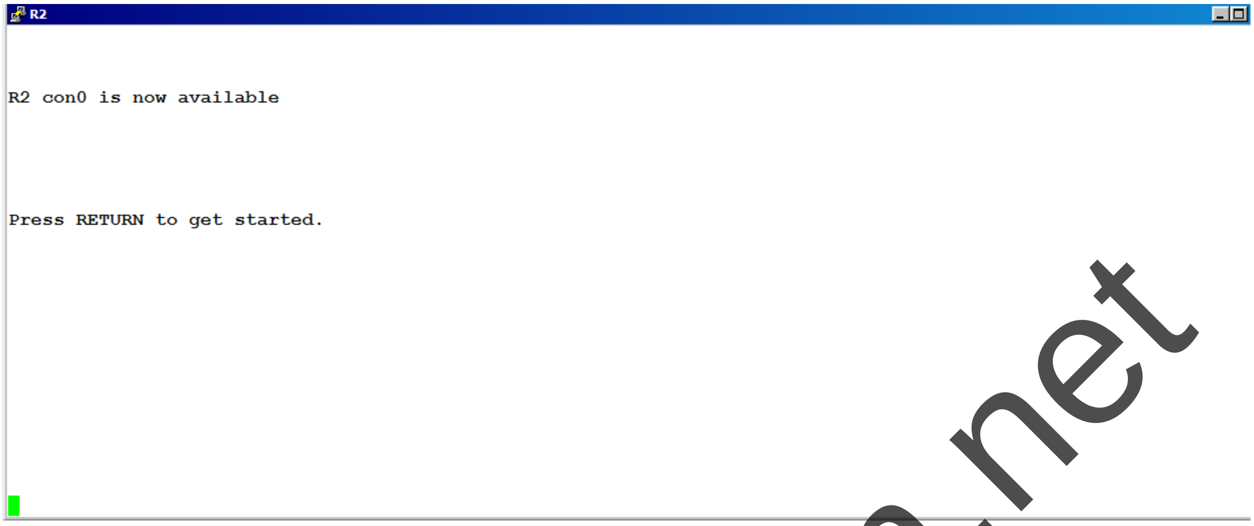
In Example 15-2, the console and auxiliary (aux) port are both configured to time out after a 5-minute interval.

Cofiguration:

```
line con 0
 exec-timeout 5 0
 privilege level 15
 password 7 00171B0F084B0A
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 5 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
```

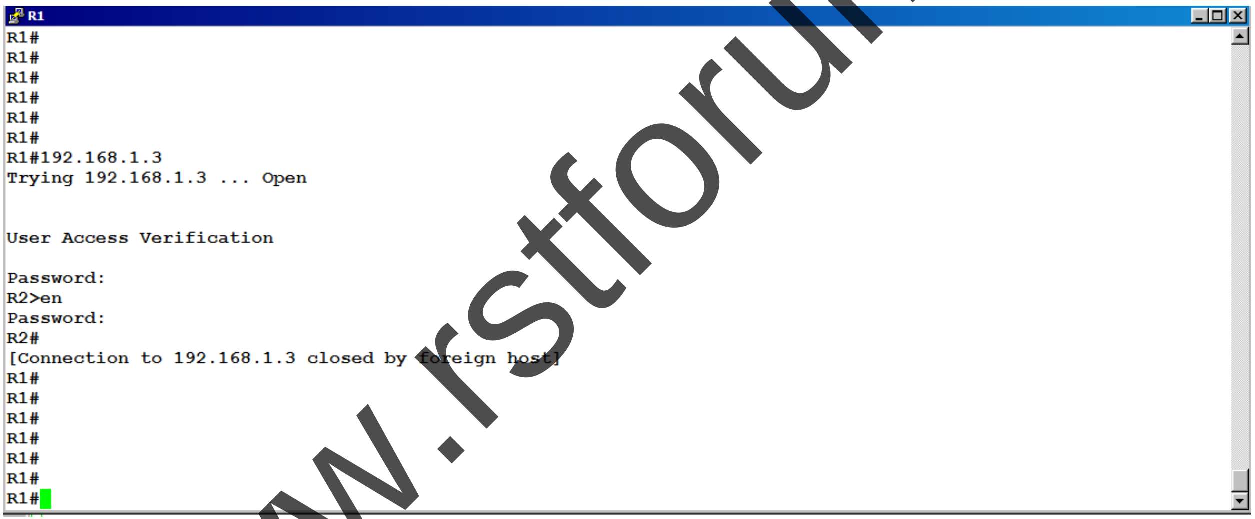
Verification:

By Console:



```
R2 con0 is now available



Press RETURN to get started.
```

By Telnet:

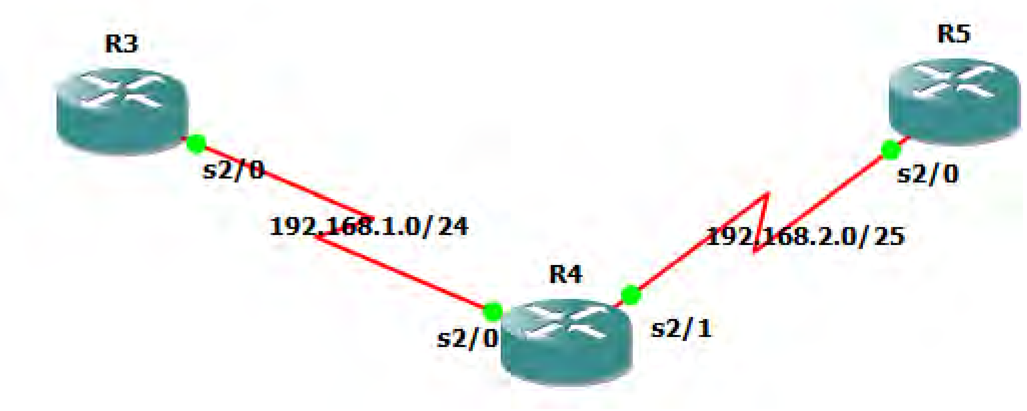


```
R1#
R1#
R1#
R1#
R1#
R1#
R1#192.168.1.3
Trying 192.168.1.3 ... Open


User Access Verification

Password:
R2>en
Password:
R2#
[Connection to 192.168.1.3 closed by foreign host]
R1#
R1#
R1#
R1#
R1#
R1#
R1#
```

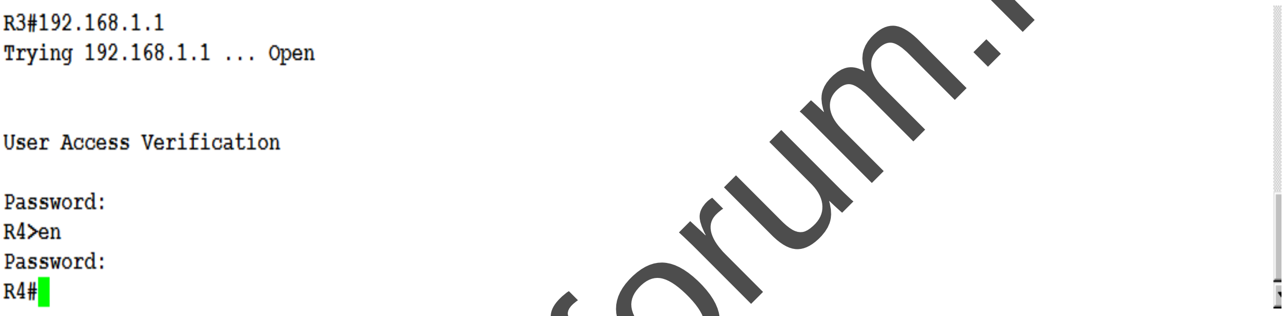
NOTE

When you are in a lab-testing environment, a constant timeout can turn into a nuisance. If security is not an immediate concern, you can choose to set the timeout interval to infinity by using the **exec-timeout 0 0** command. However, you should never do so in real-world networking.

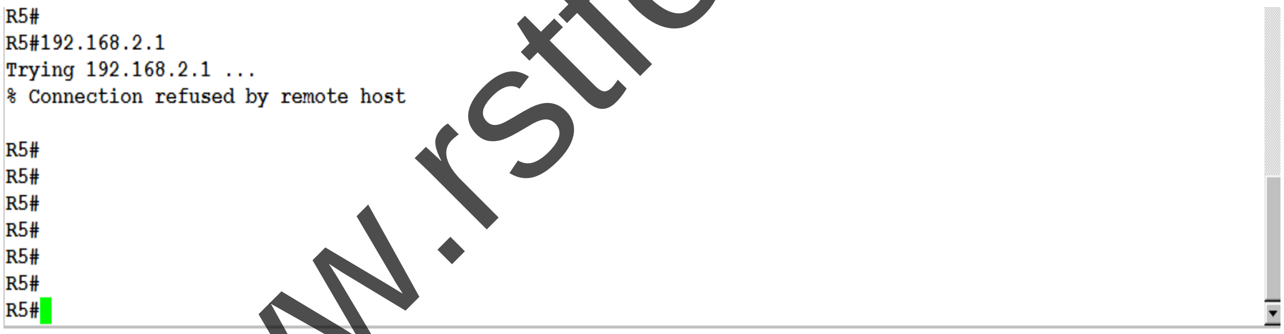
Step 3: Configuring vtys and Accessing the Network Remotely



Verification:



```
R3#192.168.1.1
Trying 192.168.1.1 ... Open


User Access Verification

Password:
R4>en
Password:
R4#
```



```
R5#
R5#192.168.2.1
Trying 192.168.2.1 ...
% Connection refused by remote host

R5#
R5#
R5#
R5#
R5#
R5#
R5#
```

As you know, vtys are used for remote network connections to the router. Generally, all the router's vtys have the same configuration. If there are extra vtys that are not used, it is a good practice to disable them with the **no line vty** command.

Applying an access list to vtys can effectively limit access to the router by specifying which connections are allowed. The command for assigning an access list to vtys is as follows:

```
R3(config-line)#access-class access-list in
```

Some of the protocols supported by the vtys (for example, rlogin and web) are not secure. To minimize the security risk, you can confine the acceptable type of connection to Telnet only with the following command:

```
R3(config-line)#transport input [telnet]
```

Example 15-3 shows IP access-list 5, which permits host 192.168.1.3. Applying access-list 5 to vty lines for inbound connections means that only one particular host can Telnet to R3 and 19.168.2.3 is not able telnet to R3. Same way R3 is able to telnet to the R2 but not R4 because access policy for telnet on R2 and R4 are configure that way so that R3 only allow to telnet to the R2 and not to the R4.

```
!
!
logging alarm informational
access-list 5 deny   192.168.2.3
access-list 5 permit 192.168.1.3
!
!
line con 0
exec-timeout 5 0
 privilege level 15
 password 7 00171B0F084B0A
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 5 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 3
 login
line vty 4
 access-class 5 in
 login
 transport input telnet
!
```

NOTE

While configuring these commands, make sure that you are connected via an aux or console port. If you perform the commands while logged in to the router via Telnet, you might inadvertently disconnect yourself.

Step 4: Creating User Accounts

In this scenario, administrators log in according to the local router database. Each administrator receives his own username, password, and privilege level assigned, which indicates the level of control an administrator has over the router. The following command places a user in a local database:

```
R3(config)#username name privilege level password string
```

In Example 15-4, five administrators are assigned to the database. When they attempt to log in, they are authenticated by their username and corresponding password and are authorized to operate on the prescribed level.

Example 15-4 Creating a Local Database

```
!

!

username shilpa privilege 3 password 7 14041A0200142B

username ssm privilege 10 password 7 08325F43

username rst privilege 10 password 7 00160012

!

!
```

Example 15-5 Designating a Privilege Level

```
!

privilege exec level 5 telnet

privilege exec level 9 enable

privilege exec level 10 disable

privilege exec level 7 show ip route

privilege exec level 7 show ip

privilege exec level 3 show startup-config

privilege exec level 7 show

!
```

Now that you have specified privilege levels for your users, you can assign a set of commands to a privilege level. Every user at the same privilege level can execute the same set. By default, every command in the Cisco IOS Software is designated for either level 1 or level 15. Level 0 exists, but it is rarely used. It includes following five commands:

- **disable**
- **enable**
- **exit**
- **help**
- **logout**

To change the default level and sign up certain commands to another level, use the following command:

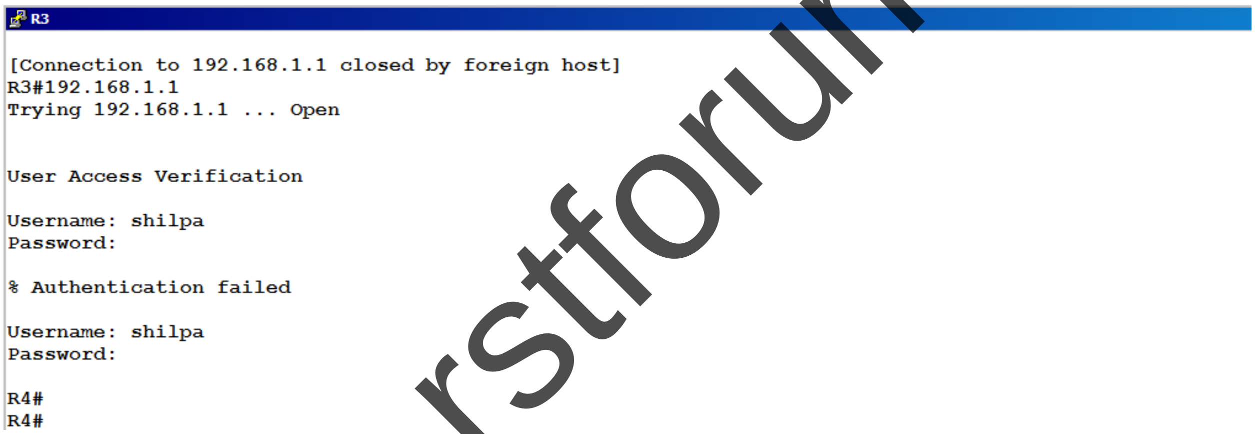
```
R8(config)#privilege exec levellevelavailable-command
```

Keep in mind that for security reasons, you should move some commands that allow too much freedom for a lower level to a higher level, not the other way around. If you move higher-level commands, such as the **configure** command, down, you might enable a user to make unauthorized changes by letting him modify his own level to a higher one. Example 15-5 shows how privilege level 3 is limited to three commands:

- **telnet**
- **show ip route**
- **show startup**

```
!

!

username shilpa privilege 3 password 7 14041A0200142B

username ssm privilege 10 password 7 08325F43

username rst privilege 10 password 7 00160012

username sam privilege 15 password 7 14141B180F0B

!

!
```

Verification:



```
[Connection to 192.168.1.1 closed by foreign host]
R3#192.168.1.1
Trying 192.168.1.1 ... Open


User Access Verification

Username: shilpa
Password:

% Authentication failed

Username: shilpa
Password:

R4#
R4#
```

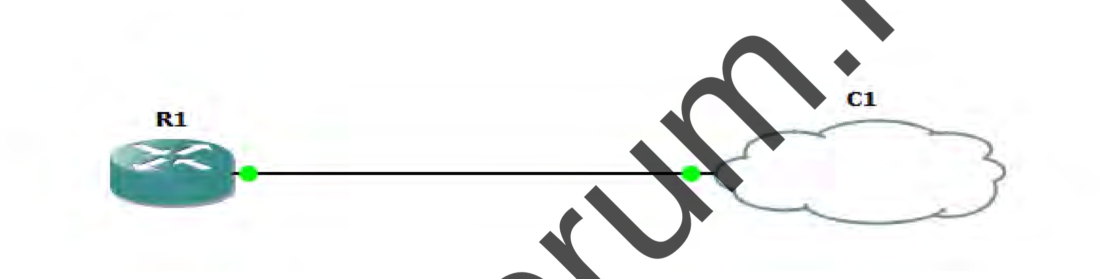
```
R5#192.168.2.1
Trying 192.168.2.1 ... Open
User Access Verification
Password:
R4>en
Translating "en"

Translating "en"
% Unknown command or computer name, or unable to find computer address
```

Step 7: Remote Administration with FTP

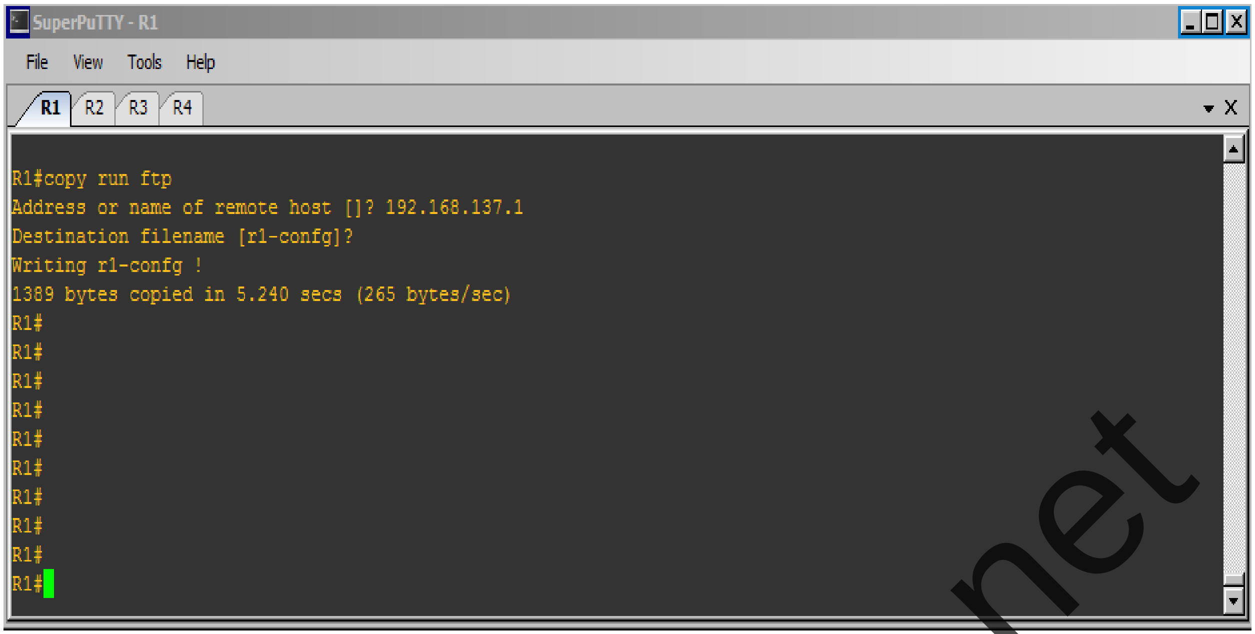
You can use File Transfer Protocol (FTP) to transfer configuration files to and from the router for remote administration. FTP is preferred because Trivial File Transfer Protocol (TFTP) does not support authentication and is, therefore, less secure and should not be used to transfer configuration files. The following commands are used to make the router FTP ready:

```
R3(config)#ip ftp source-interface interface-type number
R3(config)#ip ftp username name
R3(config)#ip ftp password string
```

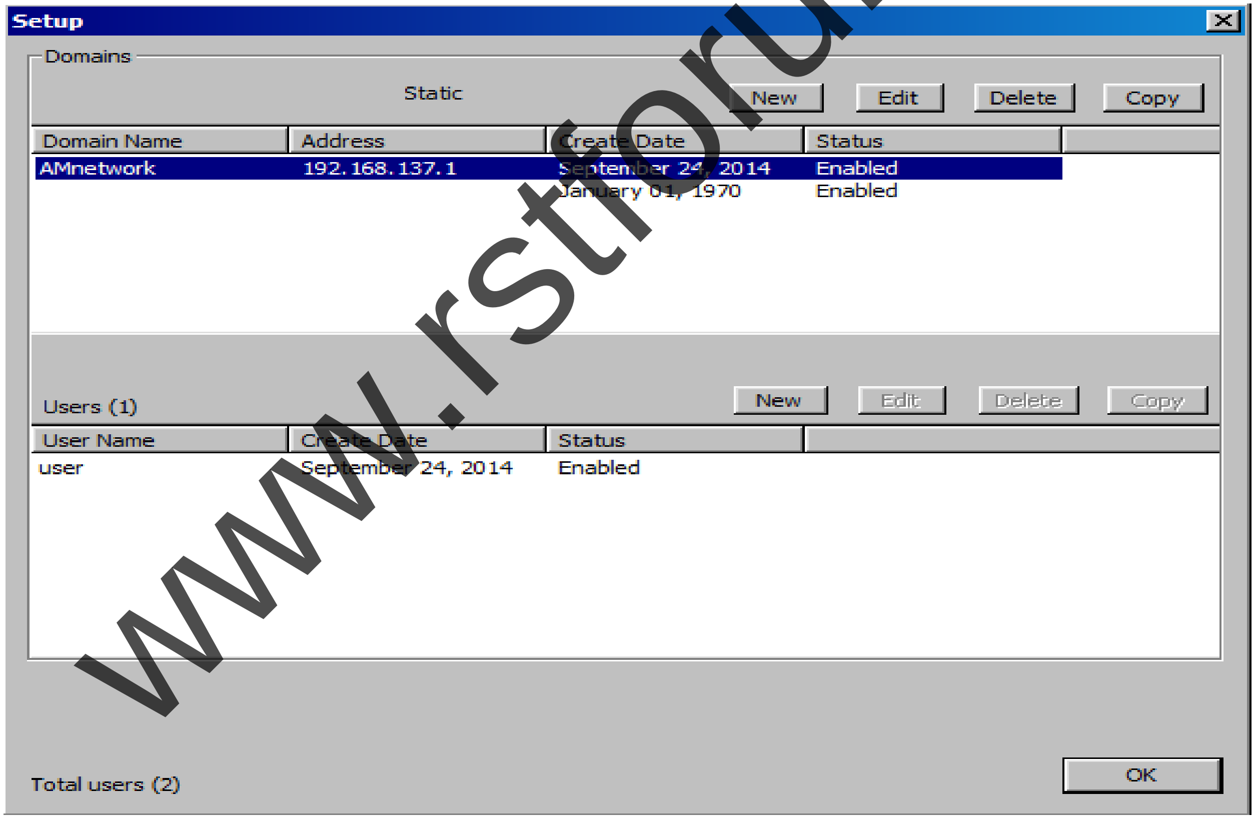
The first command specifies the local interface that is set up for the FTP connection. The two subsequent commands create the username and password for authentication on the FTP server. Example 15-7 shows the FTP configuration on R3.



```
ip ftp source-interface FastEthernet0/0
ip ftp username user
ip ftp password 7 111A0A08
no ip domain lookup
!
!
```



```
R1#copy run ftp
Address or name of remote host []? 192.168.137.1
Destination filename [r1-confg]?
Writing r1-confg !
1389 bytes copied in 5.240 secs (265 bytes/sec)
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
R1#
```

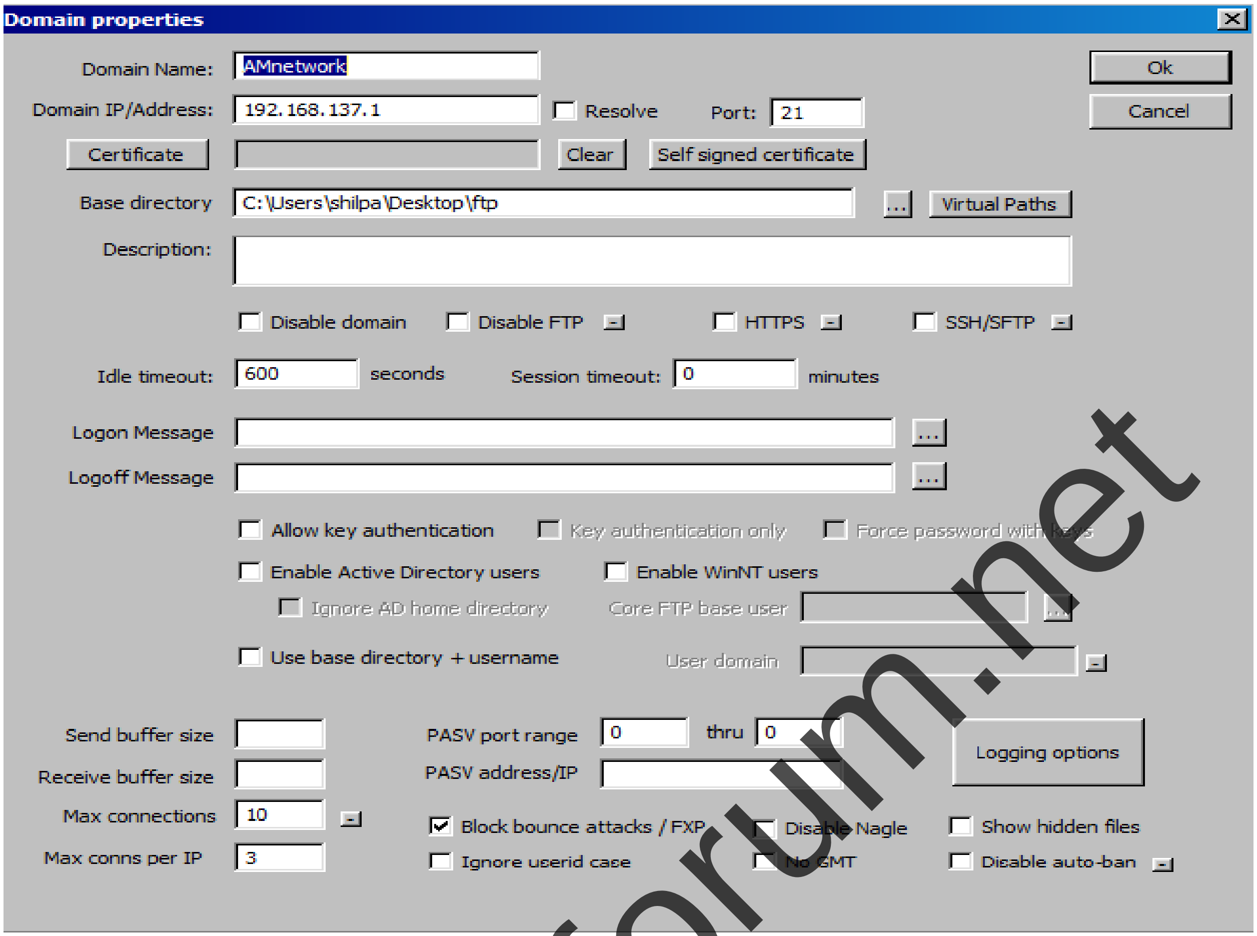


Domain Name	Address	Create Date	Status
AMnetwork	192.168.137.1	September 24, 2014	Enabled
		January 01, 1970	Enabled

Users (1)

User Name	Create Date	Status
user	September 24, 2014	Enabled

Total users (2)



Domain properties

Domain Name: AMnetwork

Domain IP/Address: 192.168.137.1 ☐ Resolve Port: 21

Certificate | Clear | Self signed certificate

Base directory C:\Users\shilpa\Desktop\ftp ... Virtual Paths

Description:

☐ Disable domain ☐ Disable FTP ☐ HTTPS ☐ SSH/SFTP

Idle timeout: 600 seconds Session timeout: 0 minutes

Logon Message

Logoff Message

☐ Allow key authentication ☐ Key authentication only ☐ Force password with keys

☐ Enable Active Directory users ☐ Enable WinNT users

☐ Ignore AD home directory Core FTP base user

☐ Use base directory + username User domain

Send buffer size

Receive buffer size

Max connections 10

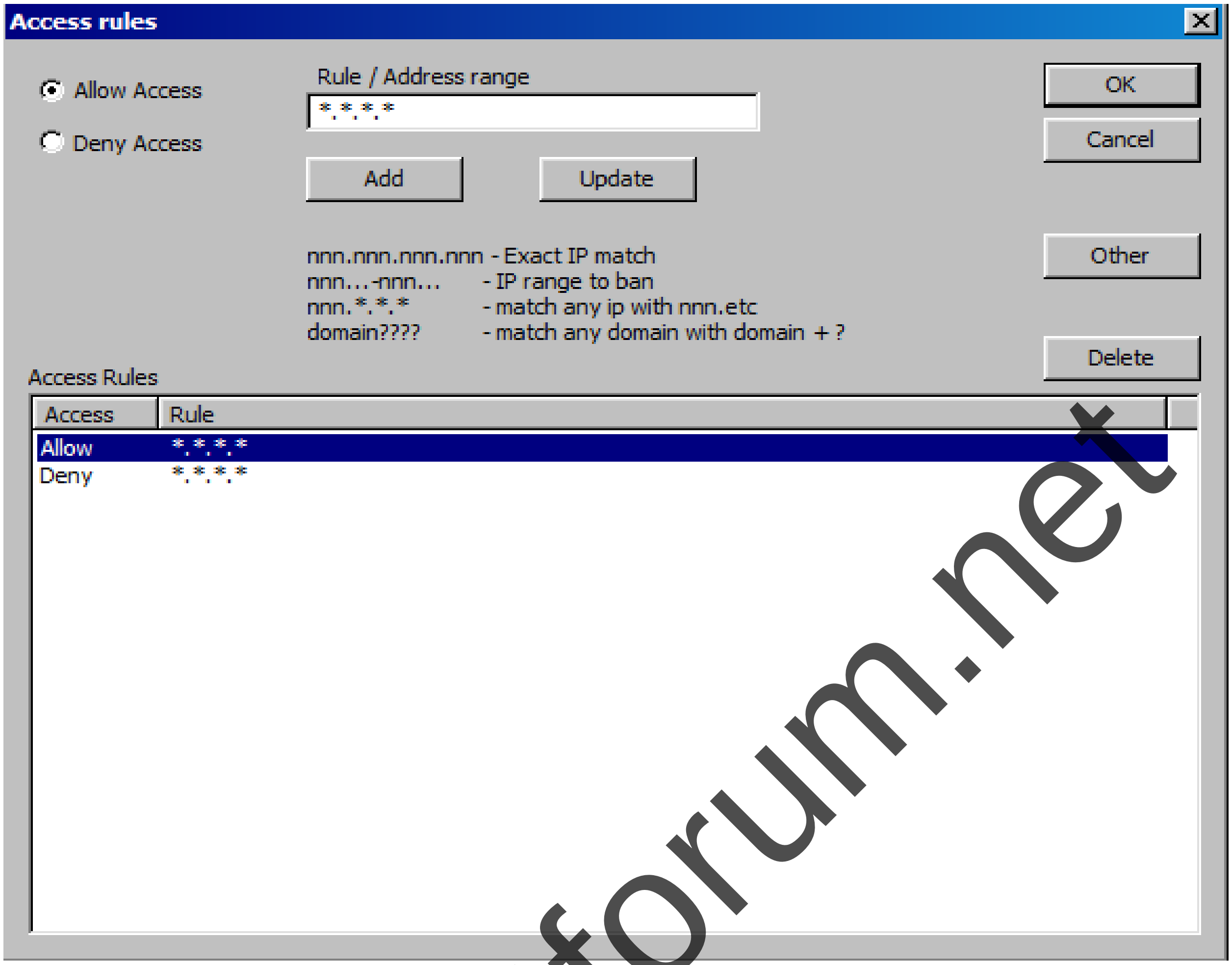
Max conns per IP 3

PASV port range 0 thru 0

PASV address/IP

☑ Block bounce attacks / FXP ☐ Disable Nagle ☐ Show hidden files

☐ Ignore userid case ☐ No GMT ☐ Disable auto-ban



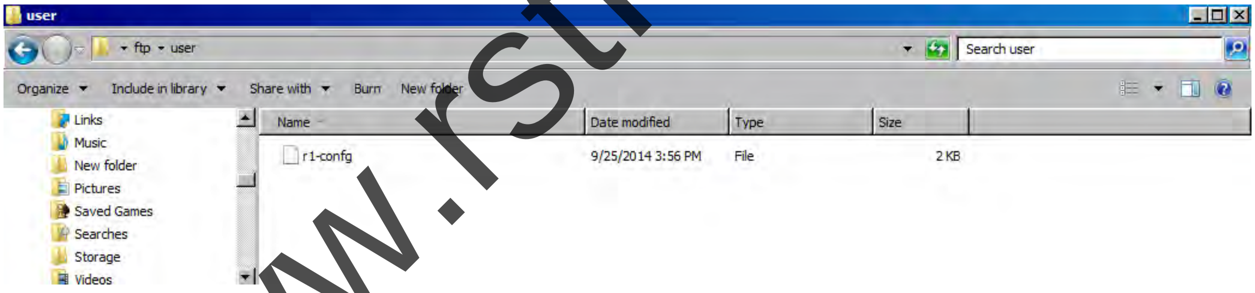
● Allow Access
○ Deny Access

Rule / Address range: *.*.*.*

```
nnn.nnn.nnn.nnn - Exact IP match
nnn...-nnn...     - IP range to ban
nnn.*.*.*         - match any ip with nnn.etc
domain????        - match any domain with domain + ?
```

Access Rules

Access	Rule
Allow	*.*.*.*
Deny	*.*.*.*



Name	Date modified	Type	Size
r1-config	9/25/2014 3:56 PM	File	2 KB

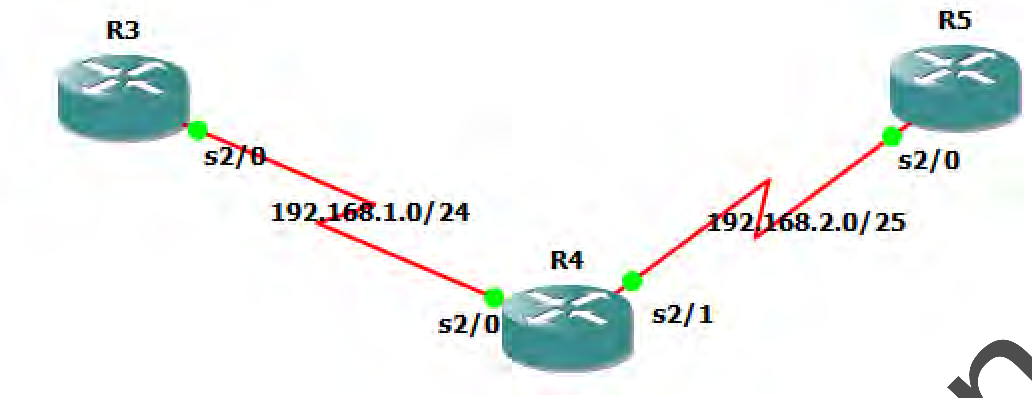
Step 8: Hiding Telnet Addresses

Normally, when you try to Telnet to a device, the router displays the address to which the connection is attempted along with other connection messages. This allows an unauthorized passerby to see it. To suppress the Telnet address, issue the following command:

```
R3(config)#service hide-telnet-address
```

Step 6: Local Authentication, Authorization, and Accounting (AAA)

Topology:



AAA has the following three separate functions:

- **Authentication**— Authentication identifies users before admitting them into a network.
- **Authorization**— Once a user is authenticated, authorization dictates what a user can accomplish on the network.
- **Accounting**— Accounting tracks the user's actions and logs them to monitor resource usage.

Example 15-6 illustrates the AAA commands configured on R3. To start an AAA process, the **aaa new-model** command is defined. The next command, **aaa authentication login default local**, names a local database as the one that is used for authentication on R3. The **aaa authorization config-commands** command enables AAA authorization of configuration commands specified by the **aaa authorization commands** statement that follows. The **aaa authorization exec default local** command specifies the local database as the source of authorization information, and the **aaa authorization commands 3 default local if-authenticated** command means that provided the user has been authenticated successfully, he is authorized by the router, after looking up the local database, to use the specified privilege level 3 commands. The latter command is helpful in the debugging process. Its practical usage is discussed in "Verification," later in this lesson.

Configuration:

User admin is authorized to operate at privilege level 3 only if the user accesses the router via vty. If the same user

```
!
aaa new-model
!
aaa authentication login default local
aaa authorization  config-commands
aaa authorization exec default local
aaa authorization commands 3 default local if-authenticated
!
aaa session-id common
!
```

NOTE

User admin is authorized to operate at privilege level 3 only if the user accesses the router via vty. If the same user attempted to access R8 via console, the user would receive privilege level 15.

Step 9: Verification

Example 15-8 demonstrates the output of the **debug aaa authentication** command followed by the **debug aaa authorization** command. The combination of these two commands shows the process a router goes through while authenticating and authorizing a user admin logging in from the remote host 192.168.1.6, permitted by access-list 5.

Example 15-8 Debugging AAA

```
R4#debug aaa authentication
AAA Authentication debugging is on
R4#debug aaa autho
R4#debug aaa authorization
AAA Authorization debugging is on
R4#
*Oct  1 16:20:35.271: AAA/BIND(0000000D): Bind i/f
*Oct  1 16:20:35.275: AAA/AUTHEN/LOGIN (0000000D): Pick method list 'default'
R4#
*Oct  1 16:20:51.123: AAA/AUTHOR (0xD): Pick method list 'default'
*Oct  1 16:20:51.131: AAA/AUTHOR/EXEC(0000000D): processing AV cmd=
*Oct  1 16:20:51.131: AAA/AUTHOR/EXEC(0000000D): processing AV priv-lvl=3
*Oct  1 16:20:51.131: AAA/AUTHOR/EXEC(0000000D): Authorization successful
R4#
*Oct  1 16:21:10.931: AAA/BIND(0000000E): Bind i/f
*Oct  1 16:21:10.939: AAA/AUTHEN/LOGIN (0000000E): Pick method list 'default'
R4#
*Oct  1 16:21:17.395: AAA/AUTHOR (0xE): Pick method list 'default'
*Oct  1 16:21:17.399: AAA/AUTHOR/EXEC(0000000E): processing AV cmd=
*Oct  1 16:21:17.403: AAA/AUTHOR/EXEC(0000000E): processing AV priv-lvl=7
*Oct  1 16:21:17.403: AAA/AUTHOR/EXEC(0000000E): Authorization successful
R4#
```

Note that the **aaa authorization config-commands** commands and **aaa authorization commands 3 default local if-authenticated** commands of this scenario's AAA configuration were not yet set at the time the **debug** commands from Example 15-8 were issued. This resulted in the debug output not displaying the user's activity after the user has been authorized.

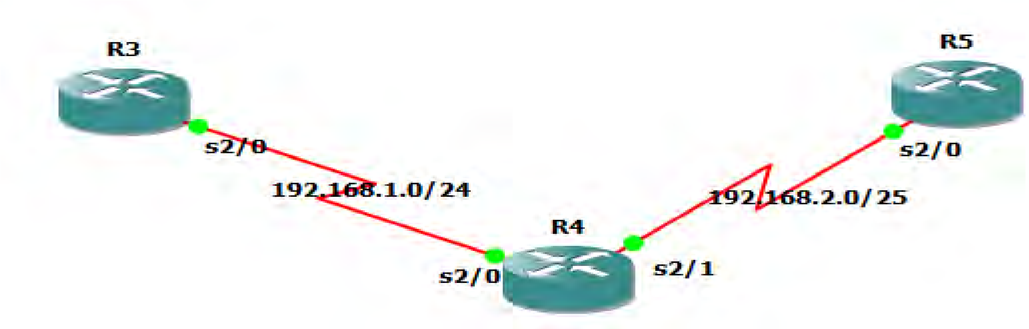
Example 15-9 shows the **debug** command output after **aaa authorization config-commands** commands and **aaa authorization commands 3 default local if-authenticated** commands have been applied. You can see that the user has issued the **show startup-config** command authorized for their privilege level.

Example 15-9 Debugging AAA after the authorization config-commands Commands

```
R4#show debug
General OS:
  AAA Authentication debugging is on
  AAA Authorization debugging is on
R4#
*Oct  1 16:29:44.403: AAA: parse name=tty3 idb type=-1 tty=-1
*Oct  1 16:29:44.403: AAA: name=tty3 flags=0x11 type=5 shelf=0 slot=0 adapter=0 port=3
channel=0
*Oct  1 16:29:44.403: AAA/MEMORY: create_user (0x65816B88) user='rst' ruser='R4' ds0=0
port='tty3' rem_addr='192.168.2.3' authen_type=ASCII service=NONE priv=3 initial_task_id='0',
vrf= (id=0)
*Oct  1 16:29:44.407: tty3 AAA/AUTHOR/CMD(3292436826): Port='tty3' list='' service=CMD
*Oct  1 16:29:44.407: AAA/AUTHOR/CMD: tty3(3292436826) user='rst'
*Oct  1 16:29:44.407: tty3 AAA/AUTHOR/CMD(3292436826): send AV service=shell
*Oct  1 16:29:44.407: tty3 AAA/AUTHOR/CMD(3292436826): send AV cmd=show
R4#
*Oct  1 16:29:44.407: tty3 AAA/AUTHOR/CMD(3292436826): send AV cmd-arg=startup-config
*Oct  1 16:29:44.411: tty3 AAA/AUTHOR/CMD(3292436826): send AV cmd-arg=<cr>
*Oct  1 16:29:44.411: tty3 AAA/AUTHOR/CMD(3292436826): found list "default"
*Oct  1 16:29:44.411: tty3 AAA/AUTHOR/CMD(3292436826): Method=LOCAL
*Oct  1 16:29:44.411: AAA/AUTHOR (3292436826): Post authorization status = PASS_ADD
*Oct  1 16:29:44.411: AAA/MEMORY: free_user (0x65816B88) user='rst' ruser='R4' port='tty3'
rem_addr='192.168.2.3' authen_type=ASCII service=NONE priv=3 vrf= (id=0)
R4#
*Oct  1 16:31:25.899: AAA/BIND(0000000F): Bind i/f
*Oct  1 16:31:25.903: AAA/AUTHEN/LOGIN (0000000F): Pick method list 'default'
R4#
*Oct  1 16:31:42.279: AAA/AUTHOR (0xF): Pick method list 'default'
*Oct  1 16:31:42.283: AAA/AUTHOR/EXEC(0000000F): processing AV cmd=
*Oct  1 16:31:42.287: AAA/AUTHOR/EXEC(0000000F): processing AV priv-lvl=7
*Oct  1 16:31:42.287: AAA/AUTHOR/EXEC(0000000F): Authorization successful
R4#
*Oct  1 16:38:21.583: AAA/BIND(00000010): Bind i/f
*Oct  1 16:38:21.587: AAA/AUTHEN/LOGIN (00000010): Pick method list 'default'
R4#
*Oct  1 16:38:32.391: AAA/AUTHEN/LOGIN (00000010): Pick method list 'default'
R4#
```

Lesson 15-2: Disabling Services

Topology:



Many services are offered by Cisco IOS Software. Although each service carries a useful function, it could present a potential security risk. When services are not used, you need to disable them. Otherwise, they open a security hole for an attacker to manipulate. This lesson is devoted to disabling unnecessary services on R3. Keep in mind that different Cisco IOS Software releases maintain different services on or off by default. If a service is off by default, disabling it does not appear in the running configuration. It is best, however, not to make any assumptions and to explicitly disable all unneeded services, even if you think they are already disabled.

The services covered in this lesson are as follows:

- Router name and DNS name resolution
- Cisco Discovery Protocol (CDP)
- TCP and UDP small servers
- Finger server
- NTP service
- BOOTP server
- Configuration auto-loading
- Proxy ARP
- IP source routing
- IP directed broadcast
- IP unreachables, redirects, and mask replies

Router Name and DNS Name Resolution

- If no Domain Name System (DNS) server is specifically mentioned in the router configuration, by default all the name queries are sent to the broadcast address of 255.255.255.255. To alter the default behavior and turn off the automatic lookup, use the following command:
- R4(config)#**no ip domain-lookup**

Cisco Discovery Protocol

- The Cisco Discovery Protocol (CDP) is a proprietary protocol that Cisco devices used to identify their directly connected neighbors. CDP is not frequently used and, like any other unnecessary local service, is considered potentially harmful to security. You can use the following commands to turn off CDP—globally and per interface:
- R4(config)#**no cdp run**
- R4(config-if)#**no cdp enable**
- Disabling CDP per interface is a nice feature because it allows you to still run CDP for the parts of the network that need it.

TCP and UDP Small Servers

Another two services that you should also turn off are the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) small servers. They are included in the list of standard TCP and UDP services that hosts should provide but are seldom needed. Use the following commands to disable TCP and UDP small servers:

```
R4(config)#no service tcp-small-servers

R4(config)#no service udp-small-servers
```

Finger Server

Next, you need to make sure that the Cisco IOS Software support for the UNIX finger protocol is disabled. Having the finger service enabled allows a user to view other active users. There are many known ways that the service can be misused and the information can fall into the wrong hands. To keep your network security in full force, you should consider turning off the finger service. After all, those who are not authorized to log in to the router have no business looking up those who do. Use the following command to disable the finger service:

```
R4(config)#no ip finger
```

NTP Service

If NTP, described earlier in "Network Time Protocol Security," is not used in the network, disable it with the following interface command:

```
R4(config-if)#ntp disable
```

BOOTP Server

In theory, BOOTP service might sound like a good idea. It is meant for use in networks where a centralized strategy of Cisco IOS Software deployment is implemented. One router can be used by other routers to load its operating system. However, the BOOTP protocol is seldom used, and it gives a hacker an opportunity to steal an IOS image. Therefore, in most situations, you should disable it using the following command:

```
R4(config)#no ip bootp server
```

Configuration Auto-Loading

The routers can find their startup configuration either in their own NVRAM or load it over the network. Obviously, loading in from elsewhere is taking a security risk. To disable the router's ability to get its configuration from the network, apply the following commands:

```
R4(config)#no boot network

R4(config)#no service config
```

Proxy ARP

Proxy Address Resolution Protocol (ARP) replies are sent to an ARP request destined for another device. When an intermediate Cisco device knows the MAC address of the destination device, it can act as a proxy. When an ARP request is destined for another Layer 3 network, a proxy ARP device extends a LAN perimeter by enabling transparent access between multiple LAN segments. This presents a security problem. An attacker can issue multiple ARP requests and use up the proxy ARP device's resources when it tries to respond to these requests in a denial-of-service (DoS) attack.

Proxy ARP is enabled on Cisco router interfaces. Disable it with the following interface command whenever it is not needed:

```
R4(config-if)#no ip proxy-arp
```

NOTE

If, however, static routes use the interface as the destination instead of a next-hop router, proxy ARP is required.

IP Source Routing

An option is found in the header of every IP packet. The Cisco IOS Software examines the option and acts accordingly. Sometimes an option indicates source routing. This means that the packet is specifying its own route. Even though it is the default, this feature has several drawbacks. First, to allow source routing in the ISP environment means that a customer selects a route as they please. Also, this feature poses a known security risk, such as a hacker taking control of a packet's route and directing it through his network. So, if source routing is not necessary in your network, you should disable it on all routers by using the following command:

```
R4(config)#no ip source-route
```

IP-Directed Broadcast

If IP directed broadcast is enabled on a router's interface, it allows the interface to respond to the Internet Control Message Protocol (ICMP) requests directed to a broadcast address of its subnet. This can cause excessive traffic and possibly bring a network down, which is a tool often used by hackers in a smurf attack.

NOTE

During a *smurf attack,* the ping requests sent to a broadcast address are forwarded to up to 255 hosts on a subnet. Because the return address of the ping request is spoofed to be the address of the attack target, all hosts that receive the ping requests reply to the attack target, flooding it with replies.

You can turn off IP directed broadcast capability on every interface with the following command:

```
R4(config-if)#no ip directed-broadcast
```

IP Unreachables, Redirects, and Mask Replies

ICMP messages that are automatically sent by Cisco routers in response to various actions can give away a lot of information, such as routes, paths, and network conditions, to an unauthorized individual. Attackers commonly use the following three types of ICMP message response features:

- **Unreachable**—A response to a nonbroadcast packet that uses an unknown protocol known as Protocol Unreachable, or a response to a packet that a responding device failed to deliver because there is no known route to a destination (Host Unreachable)
- **Redirect**—A response to a packet that notifies the sender of a better route to a destination
- **Mask Reply**—A response from a network device that knows a subnet mask for a particular subnet in an internetwork to a Mask Request message from a device that requires such knowledge

To disable the automatic messaging feature on interfaces, use the following commands:

```
R4(config-if)#no ip unreachable

R4(config-if)#no ip redirects

R4(config-if)#no ip mask-reply
```

Verification

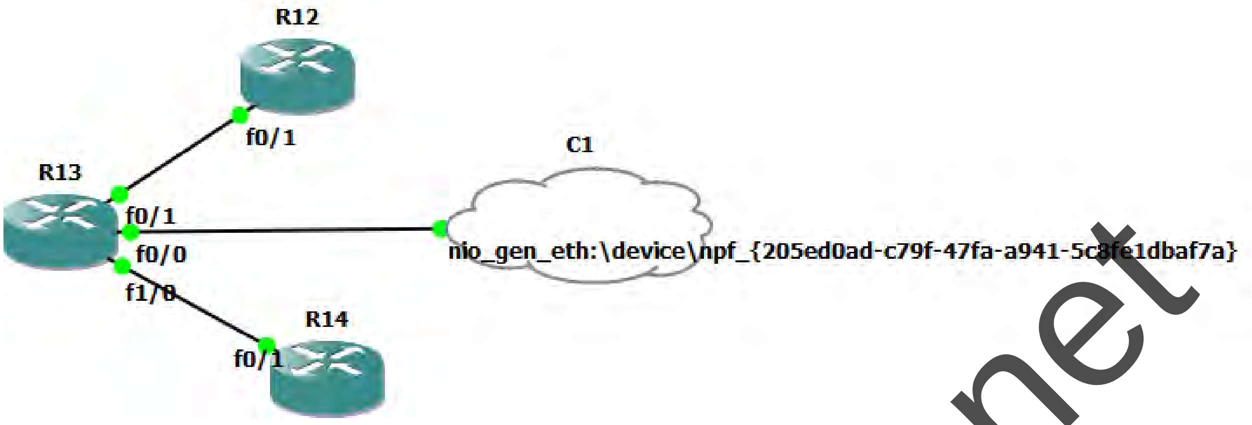
Example 15-10 shows that all the services discussed in this lesson are disabled on R8. You do not see some of them in the running configuration output because of the default settings in this particular version of Cisco IOS Software.

Example 15-10 Disabling Unnecessary Services

```
version 12.4
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname R4
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$2RAJ$mM0oAcxa6J7wQ1NmhjGar/
!
aaa new-model
!
!
aaa authentication login default local
aaa authorization config-commands
aaa authorization exec default local
aaa authorization commands 3 default local if-authenticated
!
aaa session-id common
!
resource policy
!
ip subnet-zero
no ip source-route
no ip icmp rate-limit unreachable
ip cef
ip tcp synwait-time 5
ip telnet hidden addresses
!
no ip bootp server
no ip domain lookup
!
username sam privilege 15 password 7 14141B180F0B
username admin privilege 3 password 7 045802150C2E
username joy privilege 3 password 7 01100F175804
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip address 192.168.1.1 255.255.255.0
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ntp disable
 serial restart-delay 0
 no dce-terminal-timing-enable
 no cdp enable
!
interface Serial1/1
 ip address 192.168.2.1 255.255.255.0
 no ip redirects
```

Lesson 15-3: Setting up a Secure HTTP Server

Topolgy:


R12
f0/1
C1
R13
f0/1
f0/0
f1/0
R14
f0/1
nio_gen_eth:\device\npf_{205ed0ad-c79f-47fa-a941-5c8fe1dbaf7a}

In this scenario, R4 needs to be configured as the HTTP server so that it allows remote management through the Cisco web browser interface. The syntax for the HTTP server command is as follows:

```
R13(config)#ip http server
```

Specifying the Port Number

You should change the HTTP port number from the default of 80 to something else to hide the HTTP server from an intruder. To modify the default, use the following command:

```
R13(config)#ip http port port-number
```

Specifying Authentication Technique

Next, you need to set up basic user authentication on your HTTP server. Although, you can use AAA services for this purpose, this example queries for the local database. The configuration of usernames and passwords in the database was discussed in the first lesson in "Configuring Passwords, Privileges, and Logins." Use the following command to set up basic user authentication on your local HTTP server:

```
R13(config)#ip http authentication [local]
```

Limiting Access to the Server

To limit access to the server, you can create an access list and then apply it to the HTTP configuration. To associate the list with the HTTP server access, generate the following command:

```
R13(config)#ip http access-class access-list
```

Syslog Logging

You can choose to enable the logging of a router's events to a syslog server, including the HTTP-related activity. To specify syslog logging, use the following set of commands:

```
R13(config)#logging on

R13(config)#logging facility [syslog]

R13(config)#logging source-interface local-interface

R13(config)#logging syslog-server-address

R13(config)#logging trap [alerts]
```

The first command on the list, **logging on**, turns the logging on. The **logging facility** [**syslog**] command names a syslog server as the logging monitor. The **logging source-interface** *local-interface* command identifies local interface that forwards logs to the server. The **logging** *syslog-server-address* command points to the syslog server's IP address. The **logging trap** command sets up the trap level.

Verification

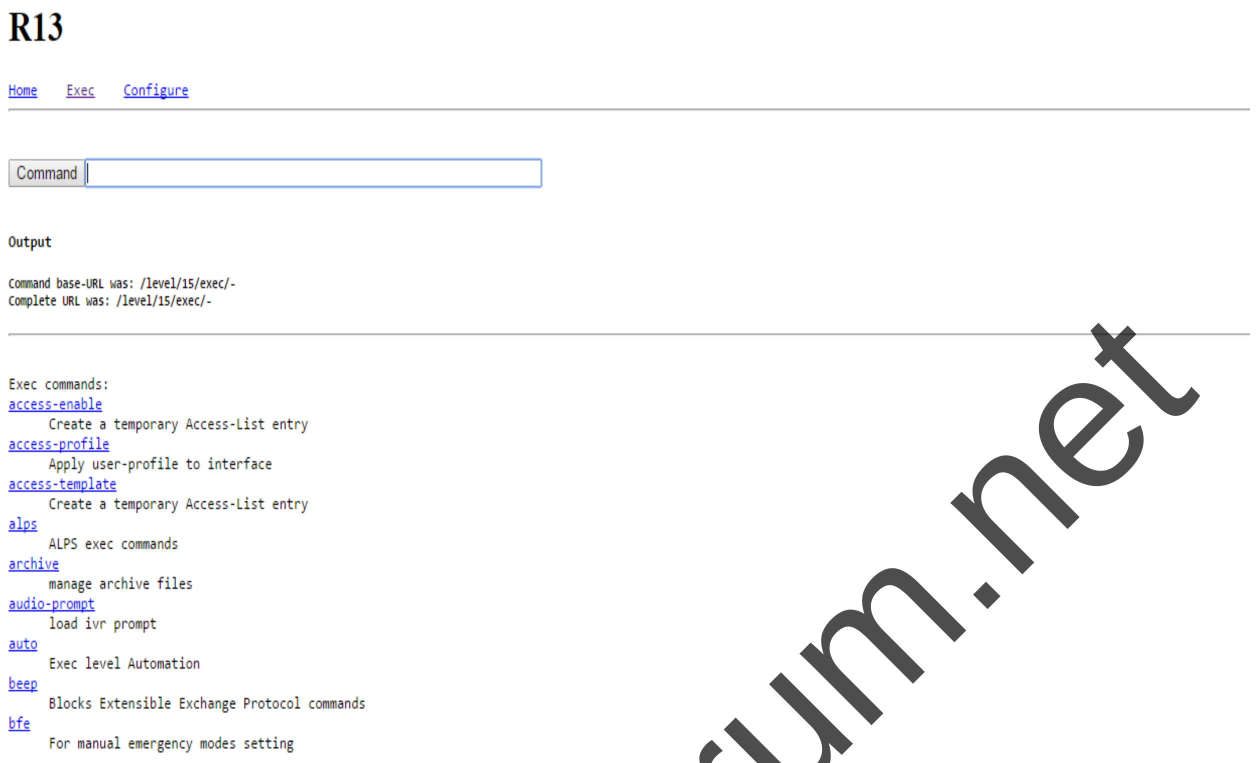
Example 15-11 displays the running configuration of R4. Notice the resolution of the HTTP commands. For example, the port number is changed to 8080. Access-list 11, permitting host 192.168.1.3 and 192.168.2.3 is a deny host, was created on R4. Serail0/1 forwards logs to the server.

Example 15-11 HTTP Configuration

```
!
ip http server
ip http port 8080
ip http access-class 1
ip http authentication local
no ip http secure-server
!
logging alarm informational
logging trap alerts
logging facility syslog
logging 192.168.1.1
logging 192.168.2.1
access-list 1 deny   192.168.2.3
!
```

```
R13#show logging
Syslog logging: enabled (12 messages dropped, 1 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)
    Console logging: level debugging, 32 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging: level debugging, 32 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (8192 bytes)
    Count and timestamp logging messages: disabled
No active filter modules.
    Trap logging: level alerts, 34 message lines logged
        Logging to 192.168.2.3(global) (udp port 514, audit disabled, link down), 0 message lines logged, xml
disabled,
               filtering disabled
        Logging to 192.168.3.3(global) (udp port 514, audit disabled, link down), 0 message lines logged, xml
disabled,
               filtering disabled

Log Buffer (8192 bytes):
sslinit fn

*Oct  2 20:40:55.267: %LINEPROTO-5-UPDOWN: Line protocol on Interface VoIP-Null0, changed state to up
*Oct  2 20:40:55.271: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Oct  2 20:40:55.275: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Oct  2 20:40:55.275: %LINEPROTO-5-UPDOWN: Line protocol on Interface IPv6-mpls, changed state to up
*Oct  2 20:40:56.087: %SYS-5-CONFIG_I: Configured from memory by console
*Oct  2 20:40:56.571: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to
down
*Oct  2 20:40:56.575: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to
down
*Oct  2 20:40:57.107: %SYS-5-RESTART: System restarted --
Cisco IOS Software, 7200 Software (C7200-ADVENTERPRISEK9-M), Version 12.4(4)T1, RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Wed 21-Dec-05 22:58 by ccai
*Oct  2 20:40:57.143: %ENTITY_ALARM-6-INFO: ASSERT INFO Fa0/0 Physical Port Administrative State Down
*Oct  2 20:40:57.147: %ENTITY_ALARM-6-INFO: ASSERT INFO Fa0/1 Physical Port Administrative State Down
*Oct  2 20:40:57.303: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
*Oct  2 20:40:57.487: %SNMP-5-COLDSTART: SNMP agent on host R13 is undergoing a cold start
*Oct  2 20:40:58.095: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down
*Oct  2 20:40:58.103: %LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down
*Oct  2 20:42:31.835: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
```

Accessing a router from browser:



R13

Home Exec Configure

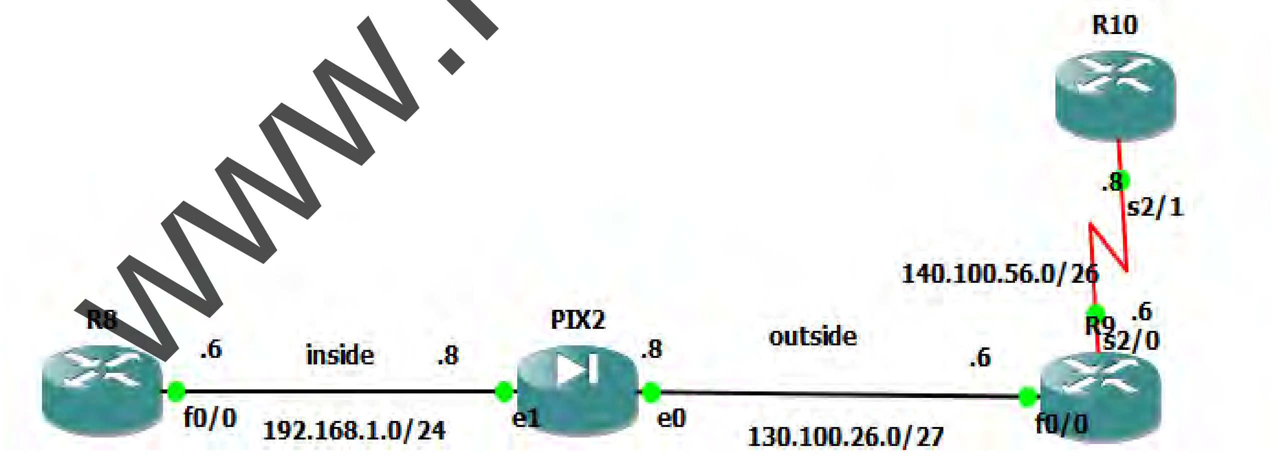
Command

```
Output

Command base-URL was: /level/15/exec/-
Complete URL was: /level/15/exec/-
```

```
Exec commands:
access-enable
      Create a temporary Access-List entry
access-profile
      Apply user-profile to interface
access-template
      Create a temporary Access-List entry
alps
      ALPS exec commands
archive
      manage archive files
audio-prompt
      load ivr prompt
auto
      Exec level Automation
beep
      Blocks Extensible Exchange Protocol commands
bfe
      For manual emergency modes setting
```

Case Study 15-1: Secure NTP Configuration

Topology:



Network Topology for NTP Configuration

Figure 15-3 describes the network topology where R6 is a client of two NTP masters: R10 and R8. To throw in a twist, PIX2 is placed between R8 and R9. This case study is not meant as an in-depth demonstration of the NTP protocol. The main goal is to achieve a functional, secure NTP configuration between the three routers using MD5 authentication.

This case study covers the following steps:

Step 1 Setting up time
Step 2 Setting up NTP relationships
Step 3 Configuring PIX2
Step 4 Restricting NTP access
Step 5 Configuring NTP authentication
Step 6 Verification

Step 1: Setting up Time

If you are using a local router as your time synchronization source, the first task you need to complete is to set the clock on the router that is to be your server, R10 in this case. The following command establishes the time (in military format) and date on the router:

```
R10#clock set hh:mm:ss day month year
```

Then, on all participating routers, set the time zone as compared to the Coordinated Universal Time (UTC). Also, configure the routers to automatically switch to daylight-saving time when appropriate. The following two commands identify the time zone and configure daylight-saving time for that zone:

```
R10(config)#clock timezone zone hours [minutes]

R10(config)#clock summer-time zone recurring

[week day month hh:mm week day month hh:mm  [offset]]
```

This scenario uses Pacific Standard Time (PST), offset 8 hours from the UTC. The summertime clock comes into effect on the first and ends on the second specified day every year, as shown in Example 15-12.

Example 15-12 Coordinating Clocks

```
R11#show run
!
Output omitted for brevity
!
Clock timezone PST -8
clock summer-time PDT recurring
```

Step 2: Setting Up NTP Relationships

When an external NTP source is not available, as is the case with this NTP configuration scenario, you need to designate a local router as the master that is to be the source of time in the network. To appoint a router as the NTP master, use the following command:

```
R11(config)#ntp master [stratum]
```

To implement redundancy, two routers act as masters: R5 and R8. When an NTP client is configured with several NTP masters, the *stratum level* of a master is the deciding factor. The stratum level of R5 is 1, and the stratum level of R8 is 3; this means that R5 takes precedence over R8.

Next, you need to set up peering between routers for clock synchronization. Use the following command:

```
R11(config)#ntp peer ip-address
```

Each router in the network has been peered up with the two other routers, as shown in Example 15-13.

Example 15-13 NTP Router Relationships

```
!

ntp master 1

ntp peer 130.100.26.8

ntp peer 140.100.56.6

!

end
```

Step 3: Configuring PIX2

Because R8 is separated from R6 by PIX2, the configuration is not fully functional without the firewall's involvement. For a comprehensive reference on the PIX functions and commands, see Chapter 23, "Cisco PIX Firewall." In this case study, you are offered a short explanation of the commands that are necessary to enable NTP between the routers.

In Example 15-14, you can see that inside and outside interfaces have been assigned their IP addresses. R6 was associated with IP address 130.100.26.6 with the **name 130.100.26.6 R6** statement. Inside-to-outside Network Address Translation (NAT) has been enabled with the **global (outside) 10 interface** and **nat (inside) 10 0.0.0.0 0.0.0.0 0 0** commands. The **static**

(inside, outside) 130.100.26.8 192.168.1.1 netmask 255.255.255.255 0 0 command specifies the outside IP address to be translated to the inside for packet forwarding to R8. The **route outside 0.0.0.0 0.0.0.0 R6 1** command designates R6 as the default gateway to the outside. Finally, the access list permitting NTP traffic destined for R8 has been applied to the inbound traffic of the outside interface.

```
!
name 130.100.26.6 R9
!
interface Ethernet0
 nameif outside
 security-level 0
 ip address 130.100.26.8 255.255.255.224
!
interface Ethernet1
 nameif inside
 security-level 100
 ip address 192.168.1.1 255.255.255.0
!
passwd 2KFQnbNIdI.2KYOU encrypted
ftp mode passive
access-list outside_access_in extended permit udp any host 130.100.26.8 eq ntp
pager lines 24
mtu outside 1500
mtu inside 1500
icmp unreachable rate-limit 1 burst-size 1
no asdm history enable
arp timeout 14400
global (outside) 10 interface
nat (inside) 10 0.0.0.0 0.0.0.0
static (inside,outside) 130.100.26.8 192.168.1.1 netmask 255.255.255.255
access-group outside_access_in in interface outside
route outside 0.0.0.0 0.0.0.0 R9 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout uauth 0:05:00 absolute
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart
telnet timeout 5
ssh timeout 5
console timeout 0
!
```

Step 4: Restricting NTP Access

You can assign an access list to the NTP process to exercise better control over your NTP synchronization. For example, R6 needs to limit the sources of its NTP updates to R5 and R8 only. To allow NTP traffic from the two routers, specify an access list, such as the one in Example 15-15, allowing 140.100.56.5 and 130.100.26.8, and apply it to NTP with the following command:

```
R6(config)#ntp access-group [query-only | serve-only | serve |
peer] access-list-number
```

```
interface FastEthernet0/0
 ip address 130.100.26.6 255.255.255.224
 ip access-group 110 in
 duplex half
 ntp broadcast
!
interface Serial2/0
 ip address 140.100.56.6 255.255.255.192
 ip ospf network point-to-point
 ntp broadcast
 serial restart-delay 0
 no dce-terminal-timing-enable
 no cdp enable
!
access-list 1 permit 140.100.56.8
access-list 1 permit 130.100.26.8
!
!
ntp access-group peer 1
ntp peer 130.100.226.8
ntp peer 140.100.56.8
!
end
```

Step 5: Configuring NTP Authentication

You have reached the final step of this configuration. NTP supports MD5 authentication, which is useful for preserving your network's security. When MD5 authentication is enforced, your router can be sure that the NTP updates that arrived are from the authorized source. To configure NTP MD5 authentication, perform the following tasks on all the participating routers:

Step 1 Start the NTP authentication process.
Step 2 Specify the NTP authentication-key, MD5 authentication type and string.
Step 3 Set up an NTP trusted key that matches the authentication-key.
Step 4 Add the authentication-key to the peer statements.

To accomplish these tasks, use the following commands and review their application on the routers shown in Example 15-16:

```
R5(config)#ntp authenticate

R5(config)#ntp authentication-key number md5 value

R5(config)#ntp trusted-key key-number

R5(config)#ntp peer ip-address [key keyid]
```

```
!

ntp authentication-key 6727 md5 070C285F4D06 7

ntp authenticate

ntp trusted-key 6727

ntp clock-period 17179922

ntp access-group peer 1

ntp master 1

ntp peer 130.100.26.8 key 6727

ntp peer 140.100.56.6 key 6727

!

end

R10#
```

Step 6: Verification

To verify that your NTP configuration is working properly, issue the following commands on any of the routers (see Example 15-17):

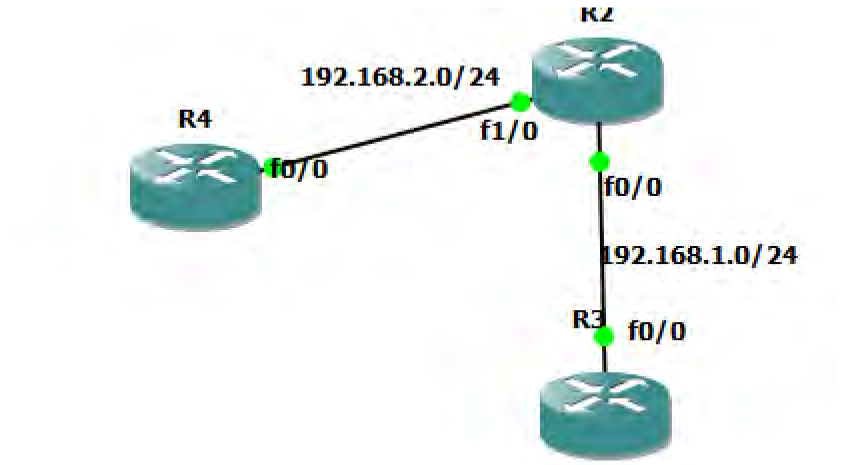
```
R5#show ntp associations

R5#show ntp status

R5#show clock
```

Example 15-17 Verifying NTP Operation

```
R10#show ntp associations

    address         ref clock      st  when  poll reach  delay  offset    disp

*~127.127.7.1      .LOCL.           0    52    64    1     0.0    0.00  15875.

 ~130.100.26.8     0.0.0.0         16     -    64    0     0.0    0.00  16000.

 ~140.100.56.6     140.100.56.8    16    28    64    0    39.9   43.44  16000.

 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

R10#

R10#

R10#show ntp status

Clock is synchronized, stratum 1, reference is .LOCL.

nominal freq is 250.0000 Hz, actual freq is 249.9992 Hz, precision is 2**24

reference time is D7D9D616.179D52E7 (19:12:38.092 PDT Fri Oct 3 2014)

clock offset is 0.0000 msec, root delay is 0.00 msec

root dispersion is 7875.02 msec, peer dispersion is 7875.02 msec

R10#

R10#

R10#show clock

19:13:11.716 PDT Fri Oct 3 2014

R10#show clock

19:13:28.032 PDT Fri Oct 3 2014

R10#
```

Case Study 15-2: Configuring SSH

Topology:



To limit SSH access to a known client only, create an access list that specifies the IP address of R6. The **access-list 15 permit 140.100.56.6 log** command is a standard access list that helps achieve the desired outcome.

The syntax for the command that assigns an inbound access list to the vtys was discussed in Lesson 15-1. When applied to this scenario, it results in the following line-mode command:

```
access-class 15 in
```

Step 2: Setting Up Usernames

The next step is to create user accounts, as described in Lesson 15-1. However, instead of using AAA, a local login has been specified here, as follows:

```
R2(config-line)#login local
```

In other words, the **login local** command indicates to the router that when a user is trying to connect via SSH, the router uses the local database configured with the **username admin privilege 15 password cisco**command to authenticate the said user.

Step 3: Generating RSA Keys

For R5 to become an SSH server, it needs to get an RSA key pair. To generate a new RSA key pair for R5, use the following command:

```
R2(config)#crypto key  generate rsa
```

At the next prompt, specify **R5.cisco.com** as the name for the keys and the default of 512 bits accepted for the key modulus. By generating the RSA key pair, you automatically enabled SSH on the router. To exercise further control over your SSH, use the commands described in the next step.

Step 4: Fine-Tuning SSH

Authentication timeout is the interval, measured in seconds, that the server waits until a client responds with a password. The default and the maximum are both 120 seconds. In this configuration, the timeout stands at 60 seconds. The syntax for configuring the authentication timeout is as follows:

```
R2(config)#ip ssh timeout  seconds
```

If a user logs in incorrectly several times, the router drops the connection. The default for a uthentication attempts is 3, and the maximum is 5. In this example, the default is kept, but the syntax for the command is as follows:

```
R2(config)#ipssh authentication-retries number
```

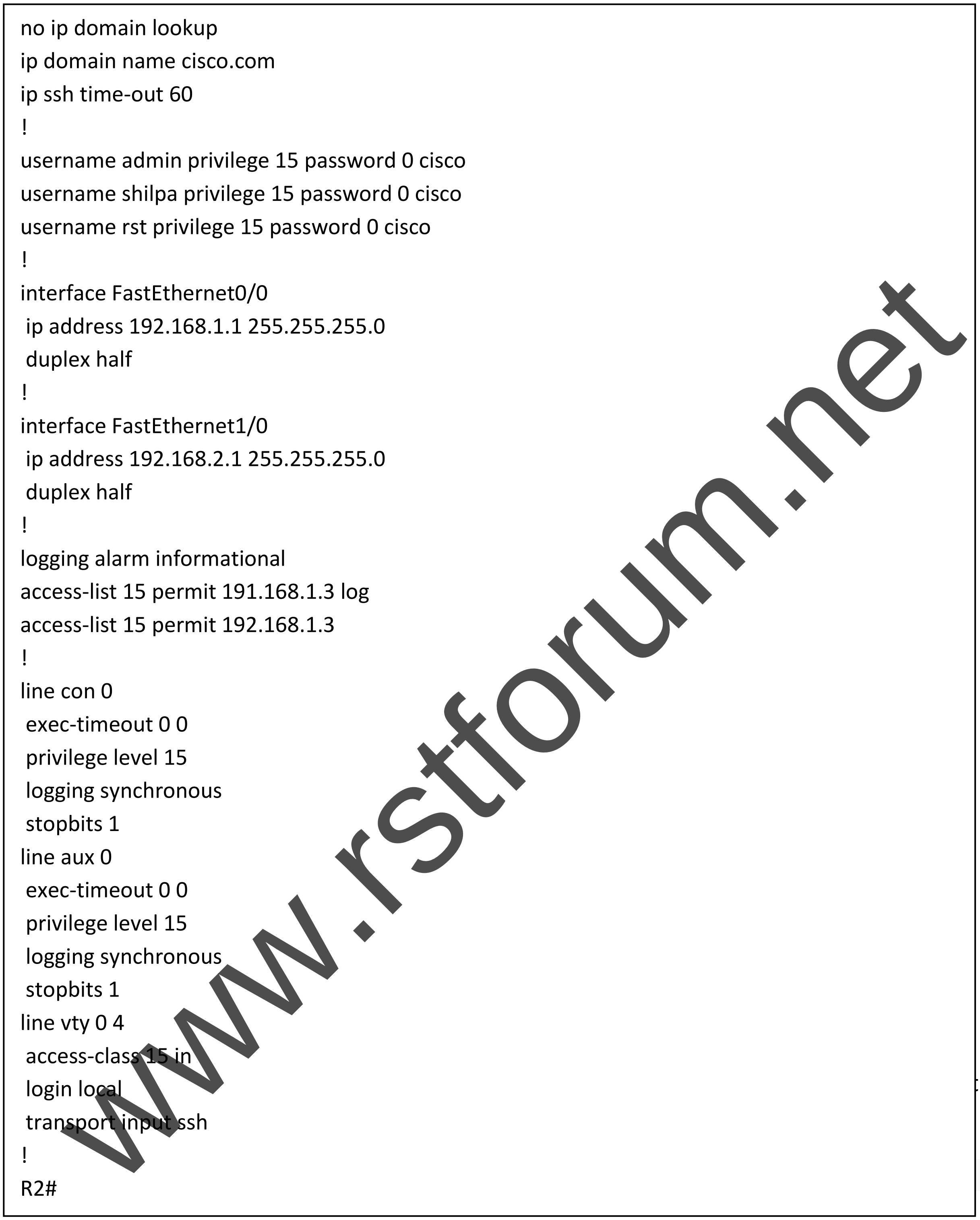
In Lesson 15-1, you allowed Telnet as the type of connection over vtys on R8. Here, you specify SSH as the connection of choice in the following manner:

```
R2(config-line)#transport input ssh
```

Step 5: Verification

Example 15-18 shows the output of the running configuration of R5. All the steps that have been covered in this case study are displayed.

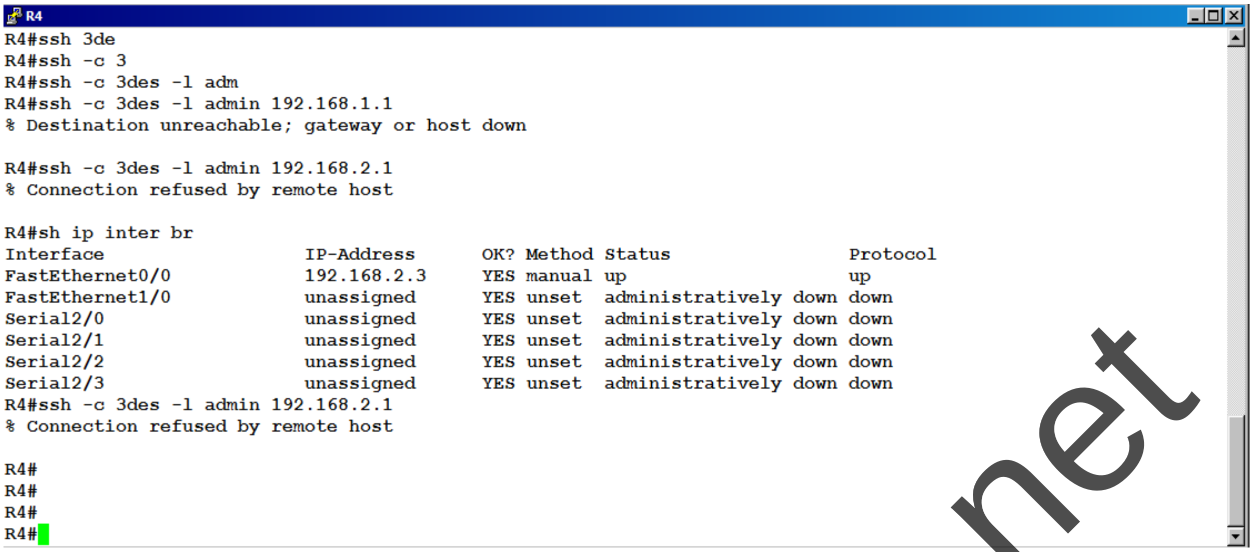
Example 15-18 SSH Configuration



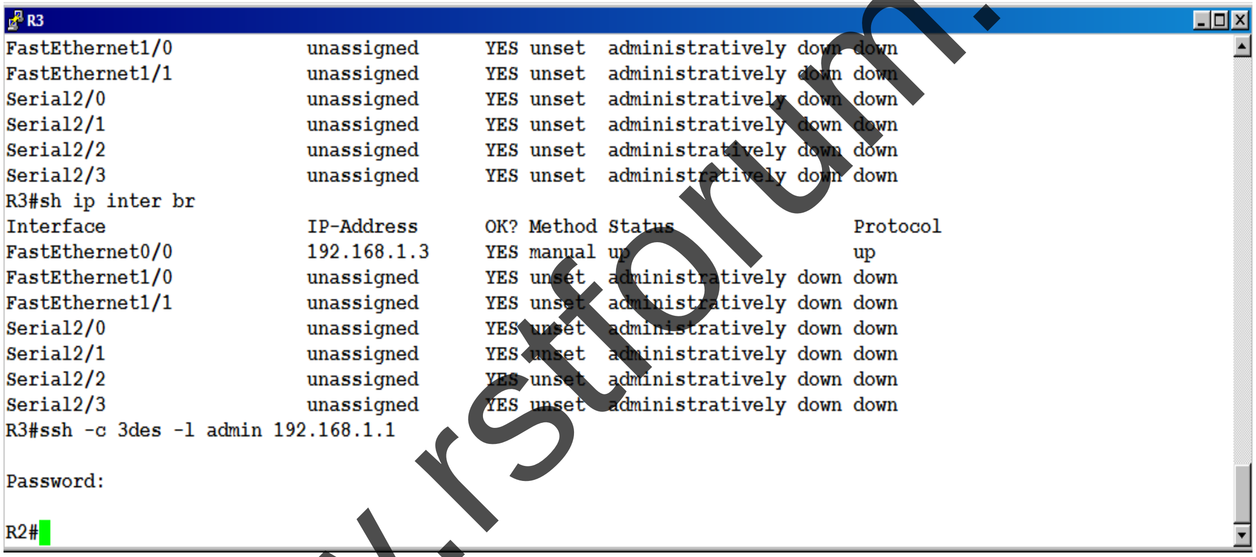
```
no ip domain lookup
ip domain name cisco.com
ip ssh time-out 60
!
username admin privilege 15 password 0 cisco
username shilpa privilege 15 password 0 cisco
username rst privilege 15 password 0 cisco
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex half
!
interface FastEthernet1/0
 ip address 192.168.2.1 255.255.255.0
 duplex half
!
logging alarm informational
access-list 15 permit 191.168.1.3 log
access-list 15 permit 192.168.1.3
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 access-class 15 in
 login local
 transport input ssh
!
R2#
```

Type in the password at the prompt.

Example 15-19 Connecting from R2 to R3 and R4 via SSH



```
R4#ssh 3de
R4#ssh -c 3
R4#ssh -c 3des -l adm
R4#ssh -c 3des -l admin 192.168.1.1
% Destination unreachable; gateway or host down

R4#ssh -c 3des -l admin 192.168.2.1
% Connection refused by remote host

R4#sh ip inter br
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.2.3     YES manual up                    up
FastEthernet1/0            unassigned      YES unset  administratively down down
Serial2/0                  unassigned      YES unset  administratively down down
Serial2/1                  unassigned      YES unset  administratively down down
Serial2/2                  unassigned      YES unset  administratively down down
Serial2/3                  unassigned      YES unset  administratively down down
R4#ssh -c 3des -l admin 192.168.2.1
% Connection refused by remote host

R4#
R4#
R4#
R4#
```



```
FastEthernet1/0            unassigned      YES unset  administratively down down
FastEthernet1/1            unassigned      YES unset  administratively down down
Serial2/0                  unassigned      YES unset  administratively down down
Serial2/1                  unassigned      YES unset  administratively down down
Serial2/2                  unassigned      YES unset  administratively down down
Serial2/3                  unassigned      YES unset  administratively down down
R3#sh ip inter br
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.1.3     YES manual up                    up
FastEthernet1/0            unassigned      YES unset  administratively down down
FastEthernet1/1            unassigned      YES unset  administratively down down
Serial2/0                  unassigned      YES unset  administratively down down
Serial2/1                  unassigned      YES unset  administratively down down
Serial2/2                  unassigned      YES unset  administratively down down
Serial2/3                  unassigned      YES unset  administratively down down
R3#ssh -c 3des -l admin 192.168.1.1

Password:

R2#
```

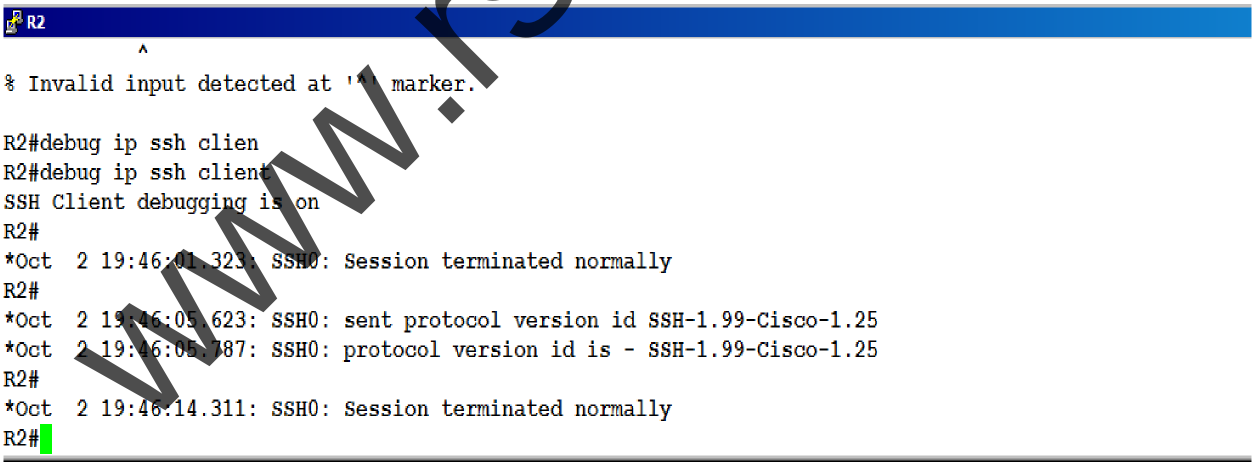
Once you are successfully connected, you can input **show ssh** on R5 to verify that SSH has been successfully enabled and check that your session is using SSH. Example 15-20 shows the output of the **show ssh** command, which displays the status of SSH server connections, and the **show ip ssh** command, which demonstrates the version and configuration data for SSH.

Example 15-20 The show ssh and show ipssh Commands on R5

```
R2#show ssh

Connection Version Mode Encryption  Hmac       State             Username

0        1.99    IN   3des-cbc    hmac-sha1    Session started      admin

0        1.99    OUT  3des-cbc    hmac-sha1    Session started      admin

%No SSHv1 server connections running.

R2#

R2#show ip ssh

SSH Enabled - version 1.99

Authentication timeout: 60 secs; Authentication retries: 3

R2#
```

If you use the Cisco IOS Software **debug ip ssh** command, you can monitor the SSH operation. Example 15-21 illustrates the output of the **debug ip ssh client** command. The first part of the output is the display of user activity, and the second is the log line that was recorded after the user exited the SSH server.

Example 15-21 The debug ip ssh client Command Output



```
           ^
% Invalid input detected at '^' marker.

R2#debug ip ssh clien
R2#debug ip ssh client
SSH Client debugging is on
R2#
*Oct  2 19:46:01.323: SSH0: Session terminated normally
R2#
*Oct  2 19:46:05.623: SSH0: sent protocol version id SSH-1.99-Cisco-1.25
*Oct  2 19:46:05.787: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
R2#
*Oct  2 19:46:14.311: SSH0: Session terminated normally
R2#
```