<CHN>CHAPTER SEVEN

<CHT>ADMINISTERING A SECURE NETWORK

<COOT>Labs included in this chapter

* <COOH1>Lab 7.1 Configuring Remote Access to Windows Server 2016
* Lab 7.2 Configuring Windows Firewall on Windows 10
* Lab 7.3 Installing and Configuring an SSH Server
* Lab 7.4 Installing and Configuring an SSH Client
* Lab 7.5 Researching IPv6

<COOBT>CompTIA Security+ Exam Objectives

<COOBL>Domain Lab

<COOB>Technologies and Tools 7.1, 7.2, 7.3, 7.4, 7.5

Architecture and Design 7.1, 7.2

# <H1>Lab 7.1 Configuring Remote Access to Windows Server 2016

**<H2>Objectives**

<TX1>Firewalls can either be hardware devices that are dedicated to performing only their packet inspection and filtering tasks, or they can be software programs installed on operating systems that have many other tasks to perform. They each have their advantages. Hardware firewalls are more secure because they don’t have to provide any other services that would open up ports and provide a larger attack surface exposure. On the other hand, these firewalls cannot inspect packets as they arrive at a host. A software firewall installed on the host computer can address threats that the hardware firewall is unable to address.

<TX2>Both Windows Server 2016 and Windows 10 contain software firewalls. In this lab, you explore the default configuration of Remote Desktop on Windows Server 2016, learn how group policies are used to control access, configure and implement Remote Desktop Protocol, and configure Windows Firewall on Windows Server 2016.

<TX2>After completing this lab, you will be able to:

* <BL>Discuss the default configuration of Remote Desktop on Windows Server 2016
* Configure and implement group policies to control access through Terminal Services
* Configure and implement Remote Desktop Protocol
* Configure Windows Firewall

**<H2>Materials Required**

<TX1>This lab requires the following:

* <BL>Windows 10 Machine with VirtualBox installed
* Windows Server 2016 VM (Firewall on)
* Windows 10 VM (Firewall on)
* Completion of Lab 6.3.

**<H2>Activity**

[BEGIN ESTIMATED COMPLETION TIME]

<FE1TX1>Estimated completion time: **15–20 minutes**

[END ESTIMATED COMPLETION TIME]

In this lab, you access the Windows Server from Windows 10 VM using Remote Desktop, and then you use the Windows Firewall to block any Remote Desktop connection attempts.

1. <NL\_FIRST>Launch the Windows Server VM and log on as Administrator.
2. <NL\_MID>If Server Manager doesn’t automatically launch then launch it. Click **Local Server**.
3. Notice that Remote Desktop is disabled. Click **Disabled** to open the System Properties dialog box.
4. Select **Allow remote connections to this computer** in the Remote Desktop section of the dialog box. See Figure 7-1.

**[Insert Figure 7-1 here]**

1. Click **Select Users**. Click **Add**.
2. Type **mtsheppard** in the Enter the object names to select text area, click **Check Names**, then click **OK** twice.
3. Click **Apply** and then click **OK**.
4. Restart the server.
5. Check to make sure that remote Desktop is enabled, by clicking **Local Server** in the Server Manager dialog. If the Server Manager did not automatically start, then start it. Note that the word “Enabled” appears next to the “Remote Desktop” heading in the properties area.
6. Now you need to verify that the firewall will allow remote access. Click **Start** and open **Control Panel**. Click **System and Security**. Click **Windows Firewall**. Click **Allow an app or feature through Windows Firewall**.
7. Scroll down to the Remote Desktop line and verify that the **Domain**, **Private,** and **Public** checkboxes are selected, as shown in Figure 7-2, and then click **OK**.

**[Insert Figure 7-2 here]**

1. Close the Allow apps dialog and open a Command Prompt window.
2. Determine your computer’s IP address by typing **ipconfig /all** and pressing **Enter**. Write down the IPv4 Address for future use.
3. Launch the Windows 10 VM. Search for **Remote Desktop**, and click the remote desktop client that appears in the list.
4. When prompted for a computer name, enter the IPv4 address of the Windows server. Enter **mtsheppard** for the user name and enter **Pa$$word** as the password.
5. When you see a warning about remote desktop usage, click **Yes**. When the desktop finishes loading, you see the Windows Server desktop. At this point, you have all the privileges that Martin Sheppard has on the server.
6. You may want to remain logged into the systems as you complete the Review Questions.

**<H2>Certification Objectives**

<TX1>Objectives for CompTIA Security+ Exam:

* <BL>2.1 Install and configure network components, both hardware- and software-based, to support organizational security.
* 2.4 Given a scenario, analyze and interpret output from security technologies.
* 2.6 Given a scenario, implement secure protocols.
* 3.2 Given a scenario, implement secure network architecture concept.

**<H2>Review Questions**

1. <FIB>Remote Desktop Protocol uses port \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
   1. <FIBA>443
   2. 22
   3. **3389**
   4. 1024
2. <MULT>Which of the following options is/are available for configuration in the Remote Desktop Connection client? (Choose all that apply.)
   1. **<MULTA>Screen size**
   2. **Local devices such as printers**
   3. Stealth mode
   4. Remote assistance
3. <TF>By default, domain administrators are members of the Remote Desktop Users group. True or **False**?
4. <MULT>Which of the following statements is correct? (Choose all that apply.)
   1. **<MULTA>Remote desktop is a program that allows Linux computers to access Windows systems using Remote Desktop Protocol.**
   2. Remote Desktop Protocol is encrypted using Secure Sockets Layer/Transport Layer Security.
   3. **The Windows Server 2016 Windows Firewall can filter incoming traffic.**
   4. On Windows Server 2016, if Remote Desktop has been enabled, users have access to Remote Desktop. Those users also have the right to log on to the server using Remote Desktop Services, but it is still necessary to manually configure the Windows Firewall to allow connections using Remote Desktop Protocol.
5. <TF>In the Windows Server 2016 Windows Firewall, an administrator can specify what computers can access the server over a particular port. **True** or False?

# <H1>Lab 7.2 Configuring Windows Firewall on Windows 10

**<H2>Objectives**

<TX1>In the previous lab, you used the Windows Server 2016 Windows Firewall to control incoming Remote Desktop Protocol packets. Windows 10 has an enhanced firewall called Windows Firewall with Advanced Security. In this lab, you use this firewall to control Web (HTTP) traffic.

<TX2>After completing this lab, you will be able to:

* <BL>Install and configure Internet Information Services on Windows 10
* Configure Windows Firewall with Advanced Security on Windows 10 to control web traffic

**<H2>Materials Required**

<TX1>This lab requires the following:

* <BL>Windows Server 2016 VM (Firewall on)
* Windows 10 VM (Firewall on)
* Completion of Lab 7.1

**<H2>Activity**

[BEGIN ESTIMATED COMPLETION TIME]

<FE1TX1>Estimated completion time: **30 minutes**

[END ESTIMATED COMPLETION TIME]

<TX1>In this lab, you install a web server on Windows 10 and then modify its properties and the Windows Firewall so that only users who know the specific port your web server is using can access your web server.

1. <NL\_FIRST>Log on to Windows 10 VM as the administrator.
2. <NL\_MID>Click Start, click Control Panel, click Programs, click Turn Windows features on or off, select **Internet Information Services**, and click OK. When the installation is complete click **Close**.
3. Launch your web browser and in the address box, type the IP address of Windows 10 VM and press Enter. If you don’t remember the IP address, open a Command Prompt window and type **ipconfig /all**. You see the welcome screen of your IIS server, as shown in Figure 7-3.

**[Insert Figure 7-3 Here]**

1. Click **Start**, then type **Notepad**. Right-click **Notepad** and choose **Run as administrator**.
2. Type the following:

<html>This is my <strong>Lab 7-2 webpage</strong>. Only chosen people will be able to access it.</html>.

1. From the File menu, click Save As, Navigate to **C:\\inetpub\wwwrooot**. Click the drop-down arrow. In the *Save as type* box, select All Files (\*.\*). In the File name box, type default.htm and click Save.
2. Return to your web browser and refresh your web server. You should now see your new webpage.
3. Launch the Windows Server VM and log on as the administrator. Open a web browser and navigate to the IP address of the Windows 10 VM.
4. Return to Windows 10 VM, click Start, click Control Panel, click **System and Security**, click Windows Firewall, and click Advanced settings in the left pane. This is the Windows Firewall with Advanced Security (see Figure 7-4). Explore this screen and then click Inbound Rules in the left pane.

**[Insert Figure 7-4 Here]**

1. In the middle pane, the firewall rules that are enabled are indicated with a green circle to the left of the rule name. The disabled rules have no check mark next to them. Scroll down to the bottom of the middle pane and double-click **World Wide Web Services (HTTP Traffic-In)**. Notice on the General tab that the Enabled box is checked and that the default action is to allow the connection.
2. Click the Programs and Services tab. No specific programs or services are subject to this rule. Click the **Remote** Computers tab. No specific computers are denied or allowed. Click the **Remote** Users tab and note that no specific users have been authorized or denied. Click the Scope tab and notice that connections are not restricted by IP address. Click the Protocols and Ports tab. Here, you see the main configuration of this rule: TCP packets that have port 80 as their destination and have any port as their source will be subject to this rule. That is, those specific packets will be allowed through the first wall. Return to the General tab, uncheck the box to the left of Enabled, and click OK. The rule now has no green circle to the left of its name.
3. Return to Windows Server and attempt to access Windows 10 VM web server, as you did in Step 8. This attempt should be successful. Close Internet Explorer.
4. Return to Windows 10 VM, click Start, click Control Panel, click the **View by** drop-down arrow, click **Small icons**, click Administrative Tools, and double-click Internet Information Services (IIS) Manager. In the left pane, expand the Computer Name node, expand Sites, and click Default Web Site. In the right pane, click Bindings. In the Site Bindings window, note that the port at which your web server is listening for HTTP requests is port 80. Web browsers assume that a website is listening at port 80; so, to make your web server more exclusive, you can change its listening port. Click the http 80 row, click Edit, and change the number in the port box to 81. Click OK and then click Close.
5. Restart IIS on Windows 10 VM by clicking on the **Computer Name** node and then, in the right pane, clicking **restart**. Return to **Windows Server** and attempt to access Windows 10 VM website using the web browser, as you did in Step 8. You can no longer do so because your browser is attempting to connect to Windows 10 VM at port 80. In your browser address bar, type Windows10VMIPAddress:81 and press Enter. This tells your browser to attempt to contact Windows 10 VM at port 81, but this, too, fails.
6. Return to Windows 10 VM and access Windows Firewall with Advanced Security. In the left pane, click Inbound Rules. In the right pane, click New Rule. In the Rule Type window, click select **Port**, and click Next. In the Protocol and Ports window, verify that **TCP** is selected. In the Specific local ports box, type 81, and then click Next. Verify that **Allow the connection** is selected and click Next. Verify that **Domain**, **Private**, and **Public** are selected and click Next. In the Name window, type Stealth Website in the Name box, type Users must know to access the web server at port 81 in the Description box, and click Finish.
7. Return to Windows Server. Access the Windows 10 VM website using the address **Windows10VMIPAddress:81** in your browser’s address box.
8. You may want to remain logged into the systems as you complete the Review Questions.

**<H2>Certification Objectives**

<TX1>Objectives for CompTIA Security+ Exam:

* <BL>2.1 Install and configure network components, both hardware- and software-based, to support organizational security.
* 2.4 Given a scenario, analyze and interpret output from security technologies.
* 2.6 Given a scenario, implement secure protocols.
* 3.2 Given a scenario, implement secure network architecture concept.

**<H2>Review Questions**

1. <FIB>By default, web servers listen for HTTP requests at port \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
   1. <FIBA>110
   2. **80**
   3. 443
   4. 53
2. <MULT>Which of the following is a parameter that can be configured for a rule in Windows Firewall with Advanced Security? (Choose all that apply.)
   1. **<MULTA>Remote port**
   2. **Protocol**
   3. **Source IP address**
   4. **Program**
3. <MULT>The Diagnose/Repair function of Windows Firewall with Advanced Security allows users to troubleshoot which of the following components? (Choose all that apply.)
   1. **<MULTA>Network adapter**
   2. **Shared folders**
   3. Web browser
   4. **Internet connections**
4. <FIB>The Main Mode and Quick Mode nodes under the Security Associations node in Windows Firewall with Advanced Security are related to the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ protocol.
   1. **<FIBA>IPsec**
   2. ICMP
   3. HTTPS
   4. HTTP
5. <TF>By default, the Remote Desktop Protocol is blocked by Windows Firewall with Advanced Security in Windows 10. **True** or False?

# <H1>Lab 7.3 Installing and Configuring an SSH Server

**<H2>Objectives**

As you learned in an earlier lab, Telnet is a terminal emulation program that passes traffic in plaintext. Although Telnet is a convenient protocol to use in configuring switches, routers, and servers, the security risks involved with passing commands, not to mention usernames and passwords, makes it too risky. Secure Shell (SSH) was created as a secure alternative to Telnet.

<TX2>SSH uses asymmetric encryption in which the two parties safely exchange encryption keys and then maintain encryption throughout the session. In this lab, you install and configure a free version of SSH called FreeSSHd.

<TX2>After completing this lab, you will be able to:

* <BL>Install and configure an SSH server

**<H2>Materials Required**

<TX1>This lab requires the following:

* <BL>Windows Server 2016 VM

**<H2>Activity**

[BEGIN ESTIMATED COMPLETION TIME]

<FE1TX1>Estimated completion time: **15 minutes**

[END ESTIMATED COMPLETION TIME]

<TX1>In this lab, you download, install, and configure a free SSH server called FreeSSHd.

1. <NL\_FIRST>Log on to Windows Server as the administrator. Open Internet Explorer and go to <URL>www.freesshd.com/?ctt=download</URL>. Click freeSSHd.exe. Save the file to your desktop.

[BEGIN NOTE]

<B1TX1>It is not unusual for websites to change where files are stored. If the suggested URL no longer functions, open a search engine such as Google and search for “freesshd.”

[END NOTE]

1. <NL\_MID>Double-click freeSSHd.exe on your desktop. In the Setup—freeSSHd SSH/Telnet Windows Server window, click Next. At the Select Destination Location window, accept the default location and click Next. At the Select Components window, accept the default of Full installation and click Next. At the Select Start Menu Folder window, accept the default and click Next. At the Select Additional Tasks window, accept the defaults and click Next and then click Install. At the Setup - Other WeOnlyDo! Products window, click Close.
2. At the Setup window, where you are prompted to create private keys, click Yes. At the Setup window, where you are prompted to run FreeSSHd as a system service, click No and then click Finish.
3. Double-click the FreeSSHd shortcut on your desktop. If necessary, click the Allow access button on the Windows Security Alert. Click OK on the thank-you message. Notice the FreeSSHd icon running in the system tray on the far-right corner of your desktop, as shown in Figure 7-5.

**[Insert Figure 7-5 Here] NOTE image is old 8-5**

1. Click the FreeSSHDService icon in the System Tray to open the freeSSHd settings window. The SSH server should be running, as indicated by a green check mark in the Server status tab. Click the Telnet tab. Notice that Telnet is not configured to start with SSHd by default. Also notice that Telnet is configured to listen at the standard Telnet port, 23.
2. Click the SSH tab. Notice that SSH listens at its default port, 22, and that the SSH server is configured to start when FreeSSHd starts. Notice the location of the cryptographic keys RSA and DSA in C:\Program Files(x86)\freeSSHd.
3. Leave the freeSSHd settings window open. Right-click the desktop, click New, Text Document, and name it SSHBanner.txt. Open the document and insert the following text: Access to this server is restricted to authorized users only. (Remember to include the period.) Save this file in C:\Program Files\freeSSHd.
4. Return to the freeSSHd settings window, click the … button to the right of the Banner message box, and browse to C:\Program Files\freeSSHd\SSHBanner.txt. Click Open. Your configuration should be similar to what is shown in Figure 7-6.

**[Insert Figure 7-6 Here]**

1. Click the Authentication tab and notice the location of the public keys and that password authentication is allowed.
2. Click the Encryption tab and note the encryption algorithms that are supported.
3. Click the Logging tab and click the box to the left of Log events to enable logging. Note the location of the log files.
4. Click the Users tab and click Add. In the User properties window, in the Login box, type administrator. In the Domain box, type Teamx.net. In the User can use section, click the box to the left of Shell. Click OK, then click OK again to close the settings box.
5. You may want to leave your stay loggedin as you answer the Review Questions.

**<H2>Certification Objectives**

<TX1>Objectives for CompTIA Security+ Exam:

* <BL>2.3 Given a scenario, troubleshoot common security issues.
* 2.6 Given a scenario, implement secure protocols.

**<H2>Review Questions**

1. <TF>FreeSSHd can listen only at a single server interface. True or **False**?
2. <MULT>Which of the following parameters can be used to determine restrictions on the use of FreeSSHd connections? (Choose all that apply.)
   1. **<MULTA>IP address**
   2. **User**
   3. Hostname
   4. **Cryptographic algorithm**
   5. **Maximum connections**
3. <MULT>Which of the following authorization types is supported by FreeSSHd? (Choose all that apply.)
   1. **<MULTA>Public key (SSH only)**
   2. Password stored as MD5 hash
   3. **NT authentication**
   4. **Password stored as SHA1 hash**
4. <FIB>SSH is considered a secure alternative to \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
   1. <FIBA>FTP
   2. Gopher
   3. **Telnet**
   4. RDP
5. <FIB>In the configuration file that FreeSSHd uses to track changes made in the freeSSHd settings window, the password for the administrator account that you created during this lab is stored as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
   1. **<FIBA>all blanks**
   2. a series of dashes
   3. the character “x”
   4. Pa$$word

# <H1>Lab 7.4 Installing and Configuring an SSH Client

**<H2>Objectives**

<TX1>In the previous lab, you created an SSH server. Now, you need to configure a client that can communicate with your server securely. PuTTY is a free SSH client that is often used in both Windows and Linux/Unix environments. In order to make an SSH connection, both sides need to negotiate the method that they will use to exchange public keys. Once this is done, the communication between the hosts is encrypted.

<TX2>Of course, to establish communication, the local firewalls need to permit SSH traffic to pass unfiltered. Leaving the SSH port 22 open is an unnecessary risk to take, especially with SSH version 1, which is vulnerable. In this lab, you configure Windows Server 2016 and **Windows 10 VM** so they can communicate using SSH.

<TX2>After completing this lab, you will be able to:

* <BL>Install and configure the SSH client PuTTY
* Implement a secure connection between two hosts using SSH

**<H2>Materials Required**

<TX1>This lab requires the following:

* <BL>Windows Server 2016
* Windows 10 VM
* Completion of Lab 7.3

**<H2>Activity**

[BEGIN ESTIMATED COMPLETION TIME]

<FE1TX1>Estimated completion time: **30 minutes**

[END ESTIMATED COMPLETION TIME]

<TX1>In this lab, you install and configure an SSH client and then implement a secure channel between hosts using SSH.

1. <NL\_FIRST>Windows Server should be configured as in Lab 7.3.
2. <NL\_MID>If necessary, log on to Windows 10 VM as the administrator. Open your web browser and go to <URL>www.chiark.greenend.org.uk/~sgtatham/putty/download.html</URL>. Scroll down to the Binaries section and click the putty.exe link. In the File Download—Security Warning window, click Save and direct the download to your desktop. Click Save. When the Download complete window opens, click **Run**.

[BEGIN NOTE]

<B1TX1>It is not unusual for websites to change where files are stored. If the suggested URL no longer functions, open a search engine such as Google and search for “PuTTY SSH client.”

[END NOTE]

1. Verify that the Connection type is set to SSH. Verify that the Port is set to 22.
2. In the left pane, click Logging. In the Session logging section, select **SSH packets and raw data**. Click the Browse button to the right of the Log file name box and select your desktop as the location for the log file. Click and explore the other entries in the Category list in the left pane.
3. Click Session in the Category list in the left pane to return to the opening screen. In the Host Name (or IP address) box, type the IP address of Windows Server and click Open. The connection attempt fails. Click OK and close the PuTTY window.
4. Log on to Windows Server as the domain administrator. Access Windows Firewall.
5. Create a new rule by clicking Advanced **settings**, Inbound Rules, and then New Rule in the Actions pane.
6. Under Rule Type, click Port, and then click Next.
7. Click Specific local ports, type 22 in the box, then click Next.
8. Click Allow the connection, then click Next. Make sure Domain, Private, and Public are checked and click **Next**.
9. Name the connection Allow SSH and click Finish.
10. Return to Windows 10 VM. Launch PuTTY again. In the Host Name (or IP address) box, type Windows Server IP Address. In the Logging window, configure the log to be stored on your desktop, as you did in Step 4. Click Open. Read the PuTTY Security Alert and then click Yes. In the PuTTY window, at the login as prompt, type administrator and press Enter. At the password prompt, type Pa$$word and press Enter.
11. You have opened a terminal session with the Windows Server through an encrypted channel. Type dir and press Enter to see the contents of the FreeSSHd directory on Windows Server. Type exit and press Enter to terminate the SSH session.
12. On your desktop, double-click putty.log. Examine the log file, which shows the packets that were exchanged during the session. See if you can identify in your log file the elements that are indicated in Figures 7-7 through 7-9.

**[Insert Figure 7-7 Here]**

**[Insert Figure 7-8 Here]**

**[Insert Figure 7-9 Here]**

1. Notice that the PuTTY log contains information that was passed during the session in plaintext (unencrypted). Launch Wireshark and repeat the connection with Windows Server using PuTTY. Examine the captured frames to determine if the transmission was successfully encrypted.
2. You may want to stay logged in as you answer the Review Questions.

**<H2>Certification Objectives**

<TX1>Objectives for CompTIA Security+ Exam:

* <BL>2.3 Given a scenario, troubleshoot common security issues.
* 2.6 Given a scenario, implement secure protocols.

**<H2>Review Questions**

1. <FIB>The Wireshark capture of the SSH session performed in this lab shows that \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
   1. <FIBA>the entire session was encrypted
   2. the entire session was unencrypted
   3. **the negotiation of cryptographic protocols was unencrypted and the rest of the session was encrypted**
   4. only the authentication password was encrypted
2. <FIB>In this lab, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ was used.
   1. <FIBA>SSHv1
   2. **SSHv2**
   3. SSHv3
   4. SSHv4
3. <FIB>In this lab, the port that Windows 10 VM used was \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
   1. <FIBA>21
   2. 80
   3. 22
   4. **dynamically assigned**
4. <TF>In this lab, after a successful connection is made, the PuTTY user can run the C:\Windows\System32\calc.exe command, which causes the calculator to run on Windows Server. **True** or False?
5. <MULT>Which of the following commands shows the ports that are used during the SSH session in this lab?
   1. **<MULTA>netstat -pn tcp**
   2. netstat -p udp
   3. ipconfig/displayports
   4. arp -a

# <H1>Lab 7.5 Researching IPv6

**<H2>Objectives**

<TX1>At one time, the depletion of IPv4 addresses seemed imminent, but the use of private IP address ranges and network address translation made it possible for IPv4 to continue to work well. However, the next generation of IP was already being created, and its developers took advantage of this opportunity to include important security features that IPv4 lacked; IPv6 includes native support for IPsec.

<TX2>The implementation of IPv6 has been slow. Although the U.S. government has converted its networks so that they support both IPv4 and IPv6, many Internet service providers have been slow to follow, and this reluctance is also found in Europe. IPv4 and IPv6 are not very compatible protocols, and migration to IPv6 is a very expensive and complicated task.

<TX2>Still, it seems likely that IPv6 will become the standard network-layer protocol in the not-too-distant future; both Windows Server 2016 and Windows 10 VM have implemented IPv6. The more you know about it, the better prepared you will be to troubleshoot network issues. In this lab, you learn about the design of IPv6 and some of its features.

<TX2>After completing this lab, you will be able to:

* <BL>Describe IPv6
* Identify IPv6 addresses
* Discuss the functional differences between IPv6 and IPv4

**<H2>Materials Required**

<TX1>This lab requires the following:

* <BL>Windows Server 2016 VM or Windows 10 VM with Internet access

**<H2>Activity**

[BEGIN ESTIMATED COMPLETION TIME]

<FE1TX1>Estimated completion time: **30 minutes**

[END ESTIMATED COMPLETION TIME]

<TX1>In this lab, you research IPv6.

1. <NL\_FIRST>Open your web browser and go to <URL>http://technet.microsoft.com/en-us/library/dd379498(v=ws.10).aspx</URL>.

[BEGIN NOTE]

<B1TX1>It is not unusual for websites to change where files are stored. If the suggested URL no longer functions, go to **technet.microsoft.com** and search for “how ipv6 works.”

[END NOTE]

1. <NL\_MID>Read **How IPv6 Works**, including the following three links: IPv6 Addressing (only through the section “Types of IPv6 Addresses”), IPv6 Neighbor Discovery, and IPv6 Routing.
2. Go to <URL>http://technet.microsoft.com/en-us/library/dd392258(v=ws.10).aspx</URL> and read the material presented.
3. You may want to leave your system logged on as you answer the Review Questions.

**<H2>Certification Objectives**

<TX2>Objectives for CompTIA Security+ Exam:

* <BL>2.6 Given a scenario, implement secure protocols.

**<H2>Review Questions**

1. <FIB>The IPv6 loopback address is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
   1. <FIBA>0000:0000:0000:0000:0000:0000:0000:0000:0001
   2. 127.0.0.1
   3. **::1**
   4. FE80:0000:0000:0000:0000:0000:0000:0001
2. <MULT>Which of the following is a valid IPv6 address?
   1. <MULTA>21DA:00D3:0000:2F3B:0000:02AA:00FF:FE28:9C5A
   2. 21DA:00D3::2F3B:02AA::9C5A
   3. **21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A**
   4. 21DA::2F3B::FE28:9C5A
3. <TF>In the IPv6 protocol, an anycast is equivalent to an IPv4 protocol broadcast. True or **False**?
4. <FIB>The IPv6 Neighbor Discovery Process performs a similar function as the IPv4 protocol \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
   1. **<FIBA>ARP**
   2. UPD
   3. TCP
   4. WINS
5. <MULT>Which of the following is a valid Netsh command?
   1. <MULTA>netsh interface ipv6> show mld
   2. netsh interface ipv6> show ipstats
   3. netsh interface ipv6> show dhcpservers
   4. **netsh interface ipv6> show joins**